

Production Linux Clusters

SC 99 Tutorial
November 14, 1999

Bill Saphir

wcsaphir@lbl.gov

Patrick Bozeman

pbozeman@lbl.gov



Rémy Evard

evard@mcs.anl.gov



Pete Beckman

beckman@acl.lanl.gov



Introduction

Brought to You By The Tribble Project

- Three leading unclassified DOE computing centers, building open tools and developing useful techniques for production Linux clusters.
 - tribble@{nersc,mcs,anl,acl.lanl}.gov
 - <http://www.nersc.gov/research/tribble>
- The LBL/NERSC PC Cluster Project
 - Plug-and-play software for small clusters
 - Scalable infrastructure for next-generation supercomputers.
 - Bill Saphir, Patrick Bozeman, Eric Roman, David Serafini
- The Argonne/MCS Chiba City Project
 - A computer science & computational science scalability testbed.
 - Development of a framework for large scale systems administration.
 - Rémy Evard, John-Paul Navarro, Dan Nurmi
- The Los Alamos/ACL Linux Cluster Project
 - Exploring Extreme Linux for computer science research and applications
 - Software tools and frameworks for programming SMP Linux clusters
 - Pete Beckman, Susan Coghlan, Ron Minnich



SC99 Tutorial Page 2



Premise/Goal

1. Clusters are cheap.
 2. People are expensive.
 3. Today's clusters are fairly good as personal supercomputers.
 4. Today's clusters are often not good as general purpose multi-user production machines.
 5. Building such a cluster requires planning and understanding design tradeoffs.
- You are here because you have heard about or experienced the mismatch of reality and expectation.
 - We will assume you know something about administering Linux systems.



Our Focus...

- | | |
|---|---|
| <ul style="list-style-type: none"> • ... is: <ul style="list-style-type: none"> • Parallel Jobs • Scalable & hierarchical • High performance • Open Source • Multiuser • High utilization • Reduce sysadmin load • Production | <ul style="list-style-type: none"> • ... is not: <ul style="list-style-type: none"> • Task farm • Mosix, single system image • PVM, HPF, DSM • Evil Empire • Small group • Dedicated resource • Grad students are cheap • Cool, it works today! |
|---|---|



The Schedule

- Morning
 - 1. Introduction
 - 2. Usage & Management Model
 - 3. Hardware Choices
 - 4. Network Choices
 - 5. Purchasing & Installation
- Lunch
- Afternoon
 - 6. Cluster Administration
 - 7. Communication
 - 8. Resource Management
 - 9. Summary
- Morning:
 - All the planning you need to do before purchasing.
- Afternoon:
 - What you do once you've got boxes of equipment in your building.
- Throughout:
 - Examples on the clusters that are here.



What Is a Cluster?

- A *cluster* is a collection of interconnected computers used as a **unified** computing resource. (Pfister)
- Clusters can offer
 - High performance
 - Large capacity
 - High availability
 - Incremental growth
- Clusters used for
 - Scientific computing
 - Making movies
 - Commercial servers (web/database/etc)



“Beowulf” Clustering

- Clustering of x86-based Linux machines for scientific computing was popularized by the **Beowulf project** at Caltech/JPL.
- “*Beowulf-class*” usually implies:
 - Off-the-shelf parts
 - Low cost LAN
 - Open source OS
 - Personal supercomputing
- From the Beowulf community we borrow an insistence on
 - Non-proprietary hardware
 - Open source software for cluster infrastructure



SC99 Tutorial Page 7



Why Linux?

- The reasons are practical, not technical.
 - Open source
 - Support for many processor families
 - Good environment for developing cluster infrastructure
 - Huge development effort means rapid improvement, support for new hardware.
 - Commercial applications
 - Talent pool
- The first three items are advantages over Windows/NT



SC99 Tutorial Page 8



Demo Clusters

- Babel
 - 12 Compaq DS10 workstations (Alpha EV6 @ 466 MHz)
 - Fast Ethernet, Gigabit Ethernet, Giganet, Myrinet
 - Rocketport serial switch
 - Diskless boot/BCCM software
- Chiba Village
 - 30 IBM Netfinity 1U workstations (500 MHz PIII x 2)
 - Fast ethernet
 - Rocketport serial switch
 - Chiba City software



SC99 Tutorial Page 9



Chapter 2: Usage and Management Model



Usage and Management Model

- Topics:
 - How do people think of programming and using these clusters?
 - How do we think of managing them?
- A few examples....



SC99 Tutorial Page 11

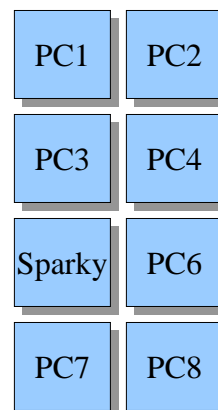


Joe's Cluster

Joe picks up a used copy of “Extremelinux” and connects a bunch of PCs in his garage.

Using Joe's Cluster:

1. Pick a PC and log in.
2. Edit & compile code.
3. Run the program.
4. Analyze results.



Usage &
Management
Model



SC99 Tutorial Page 12



Joe's Cluster - Programming

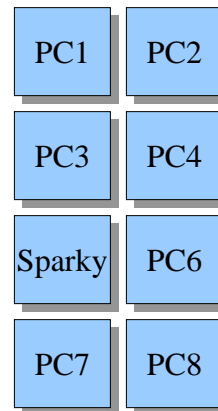
Reality sets in:

Parallel programming isn't easy. You really have to do that yourself.

Network speed matters.

There's a good reason for security patches.

Ugh. PC7 doesn't have floating point.

**MCS**

SC99 Tutorial Page 13

ACL

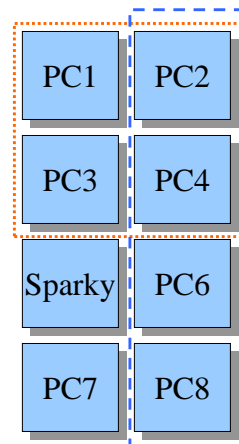
Joe's Cluster Attracts Users

Joe's neighbors want to use the cool cluster. They each only need half... (a half without PC7).

Joe's neighbors discover that using the same PC at the same time is incredibly bad.

A solution would be to use parts of the cluster exclusively for one job at a time.

And so...

**MCS**

SC99 Tutorial Page 14

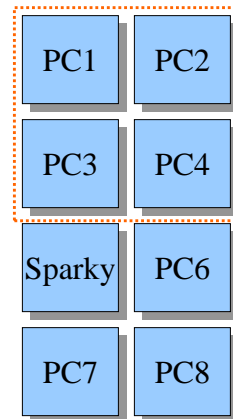
ACL

... Joe Discovers Scheduling

- Joe tries:
 - a sign up sheet
 - yelling across the yard
 - a mailing list
 - `finger schedule`
 - ...
 - a scheduler

Queue

job 2
job 3
job 4



MCS

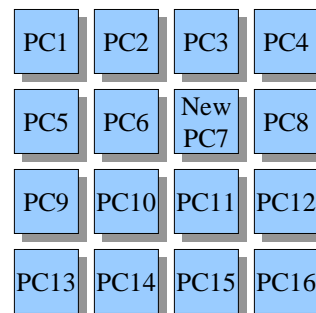
SC99 Tutorial Page 15

ACL

ComputeAtJoes.Com

- Joe expands, adding more users and more systems.
- He uses Sparky for:
 - Somewhere to login.
 - File services.
 - Scheduling services.
 - ...
- All the others are for computing.

Sparky

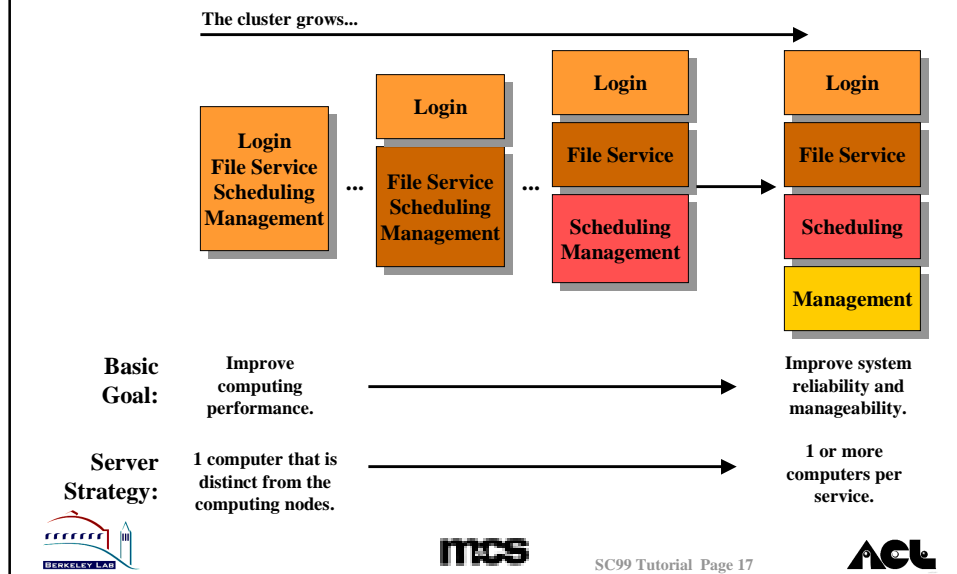


MCS

SC99 Tutorial Page 16

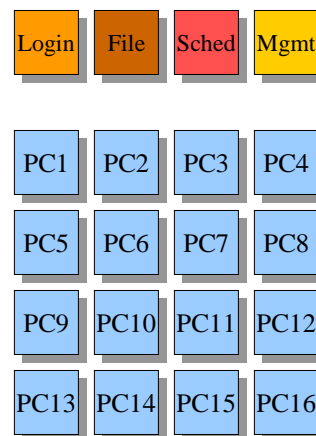
ACL

Evolution of Cluster Services



ComputeAtJoes.Com - The Franchise

- Usage Model:
 - Login to "login" node.
 - Compile and test code.
 - Schedule a test run.
 - Schedule a serious run.
 - Collect data after the run.
 - Analyze the data.
- Management Model:
 - The compute nodes are identical.
 - Run the cluster from a small set of machines.
 - Users use the login and compute nodes.



Management Approach - Ad Hoc

(The “no strategy” strategy.)

- Approach:
 - Load the OS onto each node by hand.
 - Make changes on individual nodes.
- Benefits
 - Easy.
 - Ok for small numbers of machines.
 - Natural for many sysadmins.
- Disadvantages:
 - Quickly leads to chaos and truly stunning debugging scenarios.
 - Doesn't scale.

```
%
%
%
% xtetris
```

```
dpuppy % wc -l /etc/passwd
10
```

```
aj.uf.org> wc -l /etc/passwd
52
```

```
[erwin: /]
> vi /etc/passwd
```



MCS

SC99 Tutorial Page 19



Management Approach - Clones

(The “start ‘em at the same spot” strategy.)

- Approach:
 - Have a standard OS build for your site.
 - Load that onto each node.
- Benefits
 - Fairly easy.
 - Dramatically improves consistency.
- Disadvantages:
 - Configuration drift - doesn't handle changes over time, image gets out of date.
- Tools:
 - Your own OS images and build scripts.
 - See Appendix A for Chiba Project build images, scripts, and tools.

```
%
%
% rlogin dpuppy
% vi /etc/passwd
```

Tools
build image

```
dpuppy % wc -l /etc/passwd
10
```

```
aj % wc -l /etc/passwd
10
```

```
erwin % vi /etc/passwd
```



MCS

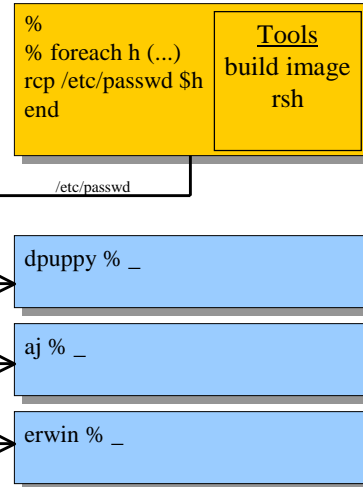
SC99 Tutorial Page 20



Management Approach - Centralize

(The “evil dictator” strategy.)

- **Approach:**
 - Use cloning.
 - Have one system from where you can drive everything.
 - Make all changes from that system.
 - Generally do a loop across all nodes, making the same change everywhere.
- **Benefits:**
 - Not too tough.
 - Improves consistency.
- **Disadvantages:**
 - Doesn't handle down nodes or new nodes.
 - Some admins will tend to hack individual nodes anyway.
- **Tools:**
 - `/rhosts` or ssh-based approach



MCS

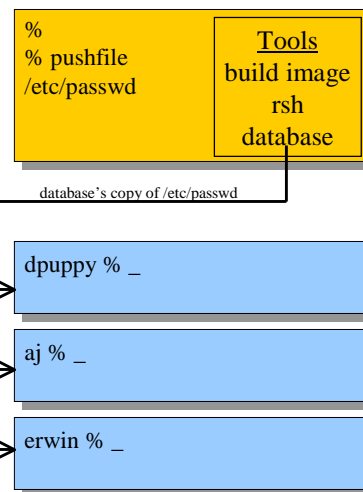
SC99 Tutorial Page 21



Management Approach - Database

(The “anal-retentive evil dictator” strategy.)

- **Approach:**
 - Use centralized management.
 - Have a database of node information which describes “how the world should be”.
 - Use scripts to get the nodes to match the database.
- **Benefits:**
 - Consistent environment.
- **Disadvantages:**
 - Requires fairly serious software infrastructure.
 - Requires rigor on part of admins.
 - Doesn't scale to very large systems.
- **Tools:**
 - chddb - mysql database of cluster info
 - hostbase - mysql database of host info



MCS

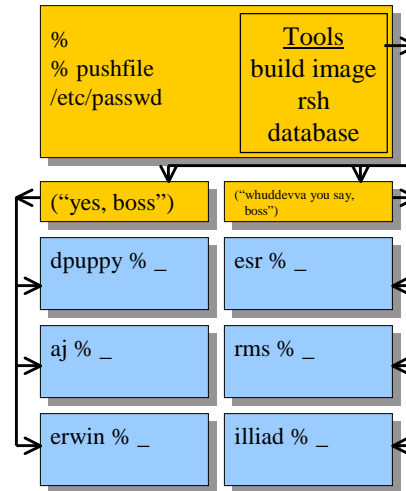
SC99 Tutorial Page 22



Management Approach - Hierarchical

(The “minions of the anal-retentive evil dictator” strategy.)

- Approach:
 - Use database management.
 - For every N nodes, have 1 system whose sole purpose in life is to make sure those nodes conform.
- Benefits:
 - Scales well to large systems.
 - Management systems can take on other responsibilities:
 - booting, file distribution, network services, ...
- Disadvantages:
 - Requires extra hardware and \$.
 - Requires fairly serious software infrastructure.
 - Breaks nice power-of-2 math.

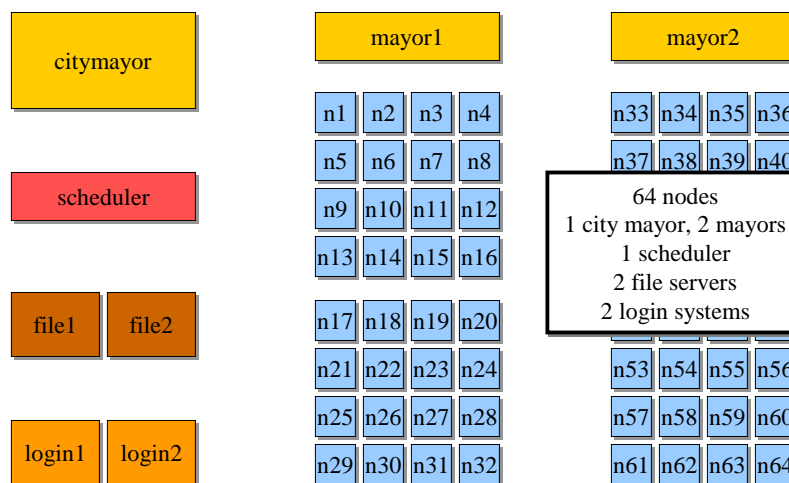


MCS

SC99 Tutorial Page 23

ACL

The Cluster We'll Use For Examples



MCS

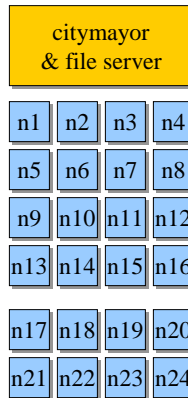
SC99 Tutorial Page 24

ACL

If Your Cluster Is Smaller.... that's cool.

- Reduce number of systems.
- Reduce management infra.
- Combine services.
- But -- the functionality doesn't go away.

scheduler



24 nodes
1 mayor/fileserver
1 scheduler
1 login



mcs

SC99 Tutorial Page 25

ACL

Fundamental Choice #1

- Management Approach
 - Ad-Hoc
 - Cloning
 - Centralize
 - Database
 - Hierarchical



mcs

SC99 Tutorial Page 26

ACL

Chapter 3: Hardware Choices

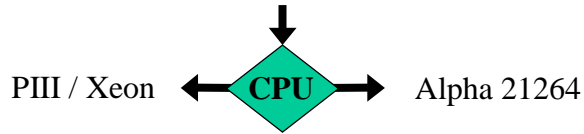


Basic Choices and Components

- Node type
 - Alpha/Pentium
 - uniprocessor/SMP
- Networking
 - TCP/IP connectivity
 - fast messaging (Myrinet, etc)
- Form factor
 - racks (yes!)
 - shelves (no)
 - unit size (1U/2U)
- Power Management
 - kludge
 - net-addressable controllers
- Basic Requirements
 - Space
 - Cooling
 - Power
- Console management:
 - BIOS/EMP/NVRAM
 - VGA
 - serial



The Heart of the System: Compute Nodes



- Performance: Xeon/500 21264/677
 - STREAM copy: 188 MB/Sec 1087 MB/Sec
 - SpecFP95: 15.1 48.4
 - Peak MFLOPS: 500 1354
 - NAS results: <http://www.nersc.gov/research/ftg/pcp/performance.html>
- Cost:
 - Dual CPU ~6K (512K L2) ~\$15K (4mb L2)
 - ~13.5K (2mb L2)
- Software: lots some



SC99 Tutorial Page 29



Compute Nodes: Other Considerations

- The motherboard implementation can be very very important:
 - PCI compatibility issues
 - Dual PCI
 - PCI performance (64 bit PCI, clock, etc)
 - SMP memory performance
 - Memory controller chipset, ECC, speed
 - BIOS / NVRAM / EMP / Power
- Integrated components:
 - 100BT, VGA
- Node style issues:
 - No local disk (net boot, NFS root)
 - No floppy, cdrom (net boot)
 - No VGA (serial console)

**Yuk, this is all
so messy!**

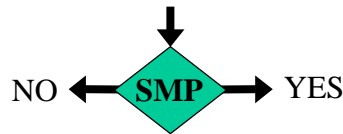


SC99 Tutorial Page 30



SMP or Uniprocessor Nodes

The obvious: Cost per CPU for a 2-way SMP cluster is less



- Disadvantages to SMP:
 - Significantly complicates many layers of software
 - Overall memory bandwidth per CPU is usually reduced
 - Memory can be more expensive (high density usually required)
- Advantages to SMP:
 - Cost of fast interconnection fabric split over 2 CPUs
 - Compact form
 - Price/Performance can be better than uniprocessor nodes (next slide)



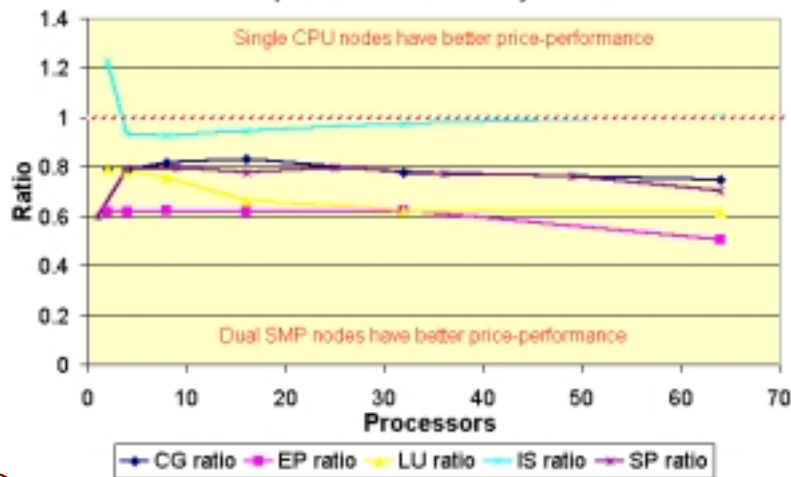
MCS

SC99 Tutorial Page 31



SMP Price/Performance

Price Performance of Single Myrinet Nodes
Compared to Dual SMP Myrinet Nodes



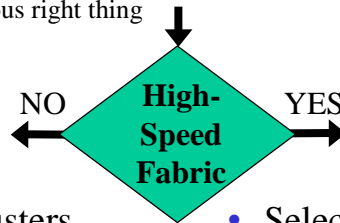
MCS

SC99 Tutorial Page 32



Networking

100BT: Just do the obvious right thing



- Build small clusters
- Do mostly embarrassingly parallel computation
- Select scalable tech:
 - Myrinet, GigE, Giganet, Quadrics, etc
- Evaluate fast MPI
 - SMP?
- Open wallet, listen to giant sucking sound.



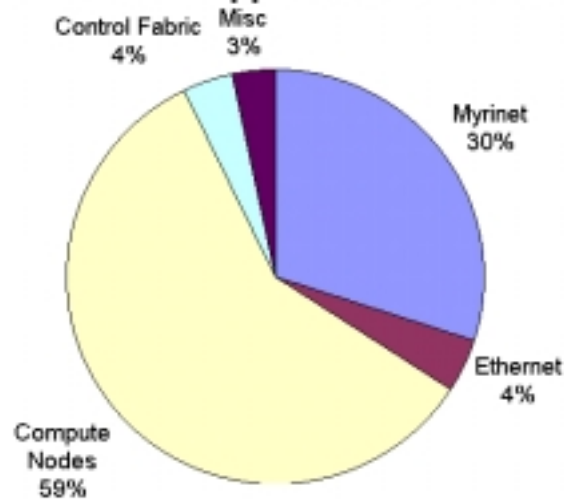
MCS

SC99 Tutorial Page 33

ACL

Bandwidth is Never Cheap

ACL: Rockhopper Hardware Costs



MCS

SC99 Tutorial Page 34

ACL

Control Fabric (1)



- Power Management Options:
 - None. Physically walk to the machine room when you need to cycle a node.
 - Save money, burn more calories, live close....
 - Use a standard, commercial net-accessible AC controller
 - BayTech <http://www.baytechdcd.com/>
 - APC <http://www.apc.com>
 - Cost: ~\$60/port
 - Compute nodes must have right “power button”
 - Bad: momentary contact button must be physically touched
 - Does On/Off switch retain state?
 - Software Utilities (MCS): power on node32

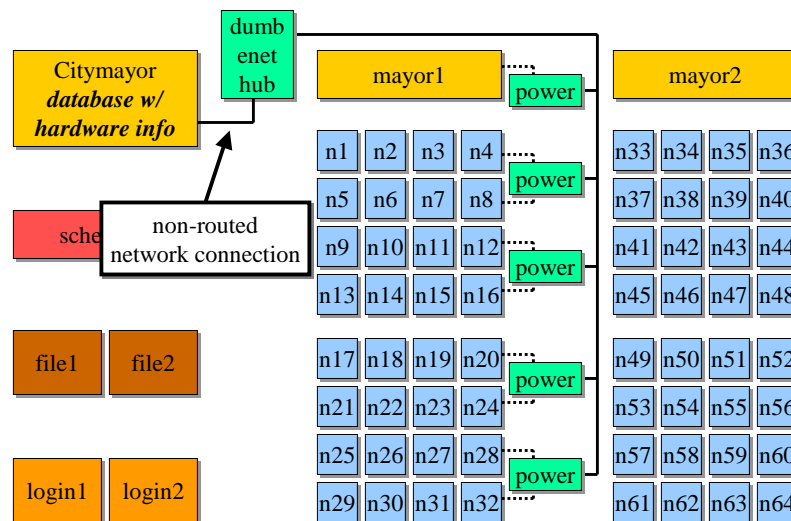


MCS

SC99 Tutorial Page 35



Power Infrastructure



MCS

SC99 Tutorial Page 36



Control Fabric (2)

- Linux Console Management:
 - Shopping cart, bungee cord, monitor, and VGA cable
 - Very small clusters can use VGA switch
 - Serial console:
 - Physically connect all the serial ports into the control node via multi-port serial card
 - Control's Rocketport: www.comtrol.com
 - Cyclades: www.cyclades.com
 - Requires software infrastructure: [conswatcher](#), [chex](#)
 - Emergency Management Port (EMP)
 - BIOS access, power/reset, NVRAM critical event log, sensor data
 - Software from VA Linux:
 - <ftp://ftp.valinux.com/pub/software/vacm/>

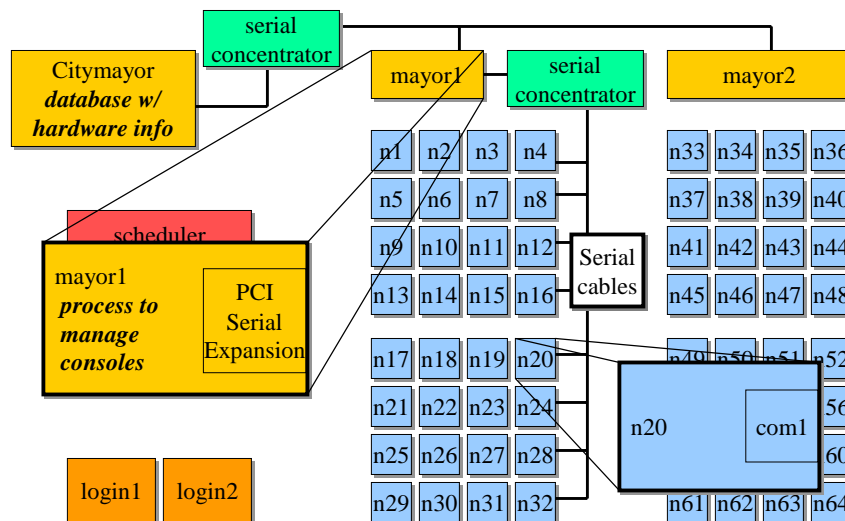


MCS

SC99 Tutorial Page 37



Serial Infrastructure



MCS

SC99 Tutorial Page 38



Concrete Example

- Scalable unit (32 compute nodes (64 CPUs), one Control Node)
 - 1 control node (Mayor)
 - Intel L440GX, PIII/500Mhz/512K, 256MB RAM, 18GB Quantum, CDROM, floppy, integrated 100BT
 - 2 32-port RocketPort serial cards
 - 4 serial port distribution panels
 - 32 compute nodes
 - Intel L440GX, Dual PIII/500Mhz/512K, 1 GBMB RAM, 9GB Quantum, floppy, integrated 100BT
 - Myrinet card
 - 2 racks
 - 1 Summit48 Enet switch with 2 GigE uplinks to PowerRail GigE switch
 - 6 BayTech power controllers (48 ports)
- Myrinet switch fabric to support all the nodes



SC99 Tutorial Page 39



The BIOS

- The BIOS controls what the machine does when it powers on.
- The Alpha BIOS is very powerful, very useful, and remotely accessible.
- The average PC BIOS is harder to work with than the average UNIX workstation BIOS:
 - No command line.
 - No remote access.
 - No OS-level access, at least not for Linux. Yet.
- BIOS configuration settings are set at the factory.
 - You might be happy with what you get.
 - You might not.
- Options for fixing the BIOS:
 - Plug in a monitor and keyboard.
 - Cope.
 - Hope that any of several Linux development projects comes through.
 - Use one of the very few BIOSs with serial support.



SC99 Tutorial Page 40



Fundamental Choices

- Management Approach - Database, Hierarchical
- Type and Number of CPUs
 - Intel vs. Alpha
 - To SMP or not to SMP
- Whether or Not to Have A Control Fabric



SC99 Tutorial Page 41



Chapter 4: Networks



What We Will Learn About Networks

- Networks have several distinct jobs
- Switching is important
- Measuring performance
- Classes of networks
- Why you should use multiple networks
- Architecture choice: how to provide external connectivity



Functions of the Network (internal)

Not all IPC is equal

- Cluster services
 - YP/DNS
 - Monitoring
 - Heartbeat
 - Process startup/management
 - Batch system
- Filesystems
 - NFS
 - Parallel filesystem
- MPI communication



Communication Patterns

- Non-parallel computing
 - Usage occurs in spurts
 - Spurts from different machines happen at random times
 - Traffic pattern is typically client-server
- Parallel computing
 - Usage occurs in spurts
 - Spurts are usually synchronized (all machines talking at once)
 - Some traffic is client/server (e.g. yp/nfs) but bulk is many to many.



SC99 Tutorial Page 45



Network Performance

- Real performance is a combination of several important factors
 - NIC, media, duplex, switches, contention, hot-spots, etc.
 - Software (TCP/IP Stack, VIA, etc)
 - CPU Utilization (benchmarks can be very misleading here)
 - System parallelization, asynchronous data movement
- There are two important network usage patterns for clusters
 - Scientific message passing for parallel programs (MPI/PVM)
 - Users want:
 - 1) low latency, high bandwidth, no network hotspots
 - 2) Asynchronous data movement, low CPU cost
 - SAN/WAN (TCP/IP)
 - Fast disk, tape, grid, home directory NetApp access



SC99 Tutorial Page 46



TCP/IP Performance

- **www.netpref.org** is the 'standard' for basic TCP/IP performance
 - ASUS P2B-F PentiumIII/500, Linux 2.2.10, Kingston KNE100-TX
 - Request/Response: 8588.43 trans/sec TCP Stream: 94.08 mbits/sec
 - SuperMicro P6DNF Dual Ppro, Linux 2.0.30pl2, 21140 Tulip (Kingston)
 - Request/Response: 4184.95 trans/sec TCP Stream: 93.58 mbits/sec
 - PPro200, Linux 2.0, Myrinet, BIP 0.9 driver
 - Request/Response: 7506.20 trans/sec TCP Stream: 338.05 mbits/sec
 - PII@450Mhz, FreeBSD 4.0, Myrinet, Duke Trapeze driver
 - Request/Response: 8535 trans/sec TCP Stream: 808 mbits/sec
- Interesting Reading: <http://www.icase.edu/coral/LinuxTCP2.html>

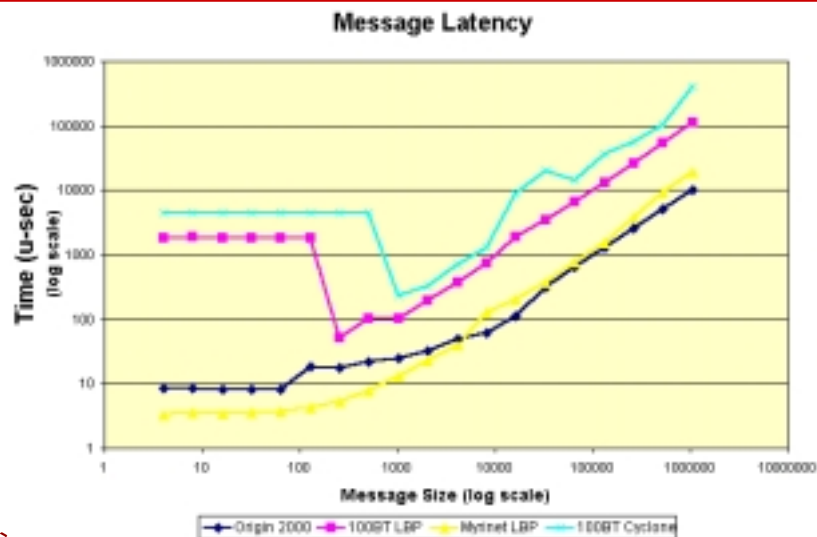


MCS

SC99 Tutorial Page 47



However, MPI over TCP/IP can be bad



MCS

SC99 Tutorial Page 48



Measuring MPI Performance

- Some Not-So-Standard Definitions:
 - **Latency**: The minimum time to get a zero-length message from A to B
 - Measurement Practices:
 - One-way latency
 - Rapid fire thousands of messages, get a single ACK
 - Half ping-pong
 - Do message ping-pongs, then divide by two
 - **Bandwidth**: The communication capacity (measured in bits/sec)
 - Measurement Practices:
 - Pre-posted Recv()
 - Report maximum for enormously large message
 - Subtract latency from timings



SC99 Tutorial Page 49



The Importance of Switching

- Parallelism in the network
 - Many-to-many can be done in parallel if network supports it. Need parallel network!
 - In worst case (e.g. FFT) $N/2$ processes sending/receiving to/from $N/2$ processes. Bisection bandwidth should be $N \times (\text{bandwidth per stream})$
 - On Fast Ethernet, switches also enable full duplex transmission
- Routing
 - MPI communication should go through switches, not nodes
 - Avoid IP routing by cluster nodes for internal traffic.
 - Fast protocols require switched-based routing

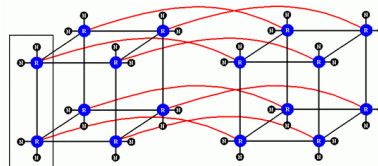
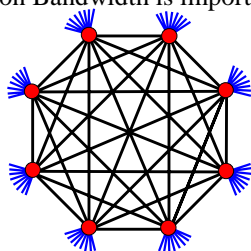


SC99 Tutorial Page 50



Bisection Bandwidth

- Bisection Bandwidth is used to report the connectivity of the entire cluster, in contrast to “latency” and “bandwidth” which are measured only between a pair of hosts.
- Theoretic Bisection Bandwidth is calculated by dividing the machine in half, using the partition of worst connectivity, and summing the bandwidth of the links between the halves.
- Interconnection topologies have “Full Bisection Bandwidth” when the number of links between any two halves of the machine is $N/2$.
- Bisection Bandwidth is important for programs that do global communication



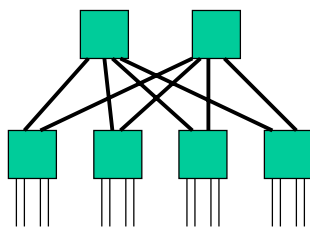
MCS

SC99 Tutorial Page 51

ACL

Big Switches

- Translating Vendor Numbers
 - Some vendors count each stream twice (once into the backplane and once out).
 - 20 Gigabit Ethernet full duplex = 10 streams * 2 directions * 2 for marketing * 1 Gb/s = 40 Gb/s for full bisection bandwidth.
- Building big switches
 - Scalable networks require switches that can be connected arbitrarily
 - Building a $2N$ -port full-bisection switch from N -port full-bisection switches requires $6N$ -port switches and $2N$ cables!

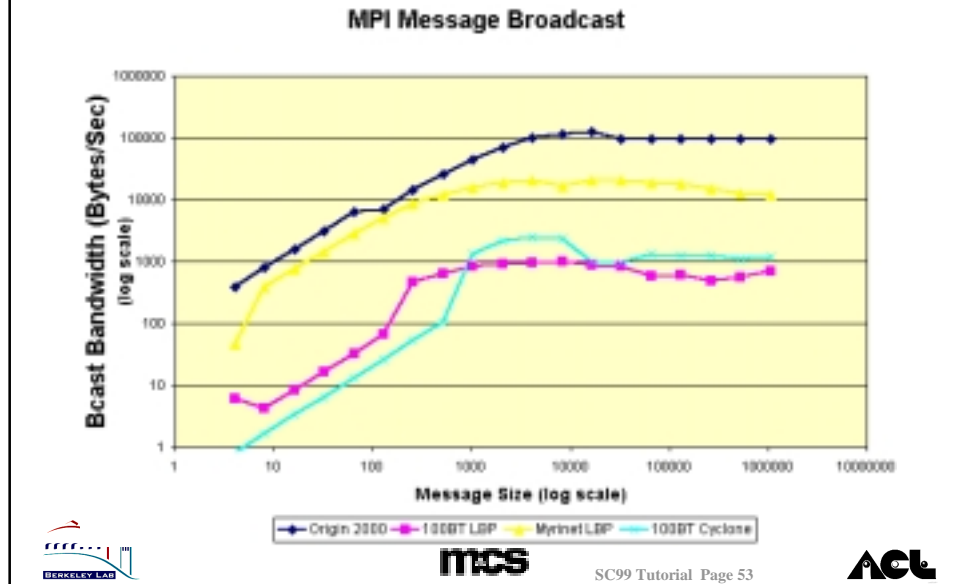


MCS

SC99 Tutorial Page 52

ACL

Example: Bisection Matters



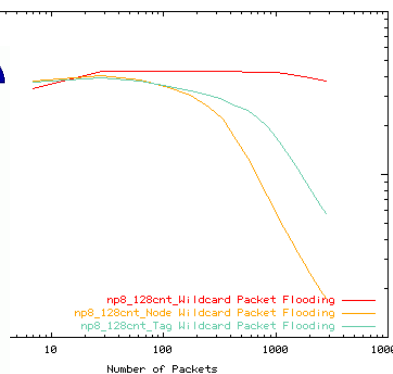
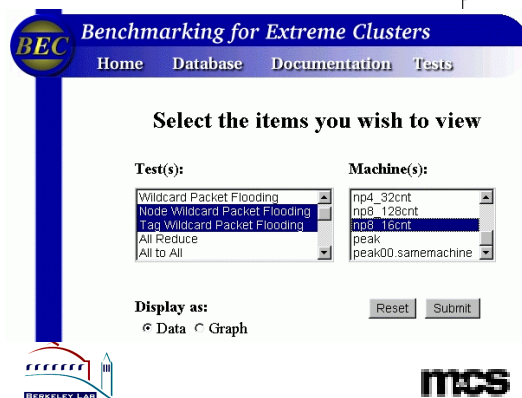
Ugh! Number Overload! **HELP!!!**

What does it all mean?

- The overall interconnection performance of your cluster is not easily characterized by two or three scalars.
- There are many ways to mislead with “latency” and “bandwidth” numbers for a given network. Run the tests yourself.
- Bisection Bandwidth is very important for some applications
- The only true measure:
 - How will my application run? How will my application scale?
- TCP/IP is generally a poor transport for MPI-style messaging
- “User-level” messaging or “OS Bypass” are techniques for improving the performance of messaging
- CPU Utilization and asynchronous data movement can be important, but remains largely unmeasured

Running the Benchmarks Yourself

- MPBench has been combined into LLCbench
 - <http://icl.cs.utk.edu/projects/llcbench/>
- Another suite is BEC: Benchmarking Extreme Clusters
 - <http://www.acl.lanl.gov/bec>



MCS

SC99 Tutorial Page 55

ACL

“Legacy” Networks

- These networks are fine for most cluster services.
 - Scaling is quite limited. A single switch can have full bisection bandwidth, but beyond that generally not possible.
- Fast Ethernet
 - 100 Mb/s; switches available up to about 100; Switches essential to support full duplex.
 - Look for about \$150 a port, total (including switch)
 - Gigabit uplink to servers is useful to avoid many-to-one problems.
- Gigabit ethernet
 - 1000 Mb/s; 30-40 MB/s in practice. Latency the same as Fast Ethernet. switches available up to about 64.
 - Has been very expensive. Now look for \$1000 a port (including switch) with new copper-based NICs/switches.



MCS

SC99 Tutorial Page 56

ACL

User-space communication

- Time spent transferring data is time not spent computing.
 - TCP has many overheads.
 - For best performance, need a **protected user-space** communication system (kernel not involved in communication).
 - User-space communication requires special hardware support.
 - Until recently: Only Myrinet; many APIs
- **Virtual Interface Architecture**
 - New industry standard; used by System I/O
 - Single API for any network.



New Gigabit Networks

- Several new networks provide
 - Support for user-space communication
 - Ability to connect switches to build arbitrarily robust networks.
 - MPP-like performance
- **Myrinet**
 - Made by Myricom, Inc.
 - >100 MB/s; soon > 200 MB/s
 - The only choice for building very large networks (large switches, proven scalability)
 - Support user-space communication software is called “gm” -- latency not wonderful. There is a basic MPI over GM . Supports TCP simultaneously.
 - Programmable processor on NIC
 - Expect to pay ~\$1500/node
 - VIA support coming soon.



New Gigabit Networks, Continued

- **Giganet**
 - Native VIA network. Does not support TCP (yet)
 - 100 MB/s; 8us latency;
 - Currently small switches only.
- **Servernet II**
 - Native VIA network. Does not support TCP (yet). (?)
 - Successor to Servernet I used in Tandem non-stop servers.
 - Available 1Q2000.
- **System I/O/InfiniBand**
 - Will be usable as a network.
 - Many implications for cluster computing.



M-VIA/MVICH

- M-VIA is an implementation of VIA for Linux
 - Supports Myrinet, Giganet, Servernet II and other networks
 - Very high performance
 - No recompilation/relinking necessary for different networks (important for code distribution)
- MVICH is an implementation of the MPICH ADI for VIA
 - Zero-copy protocols
 - Early in development cycle.
- Get M-VIA and MVICH on BLD web site.



Why You Should Use Multiple Networks

- You need a high-performance network, but:
 - Parallel programs are very sensitive to perturbations (everything is an $O(1)$ effect).
 - High performance networks tend to be fragile
 - High performance networks may not be available before routing is set up by software
- Redundancy in critical infrastructure is good. (E.g. have Ethernet as a backup even if Myrinet is your primary).
- Security -- next slides.



3 Network Configuration Issues

- Public vs. Private network addresses
- Routing via switch or host (including masquerading)
- How to configure multiple networks



Private Networks

- Three sets of network addresses are reserved for anyone's use. Just don't let packets with these addresses out of your cluster!
 - 10.0.0.0 - 10.255.255.255
 - 172.16.0.0 - 172.31.255.255
 - 192.168.0.0 - 192.168.255.255
- Security. Complete trust inside a private network is possible.
 - Firewalling/packet filtering less complicated.
 - Some tools rely on complete (rsh-style) trust.
 - Many services do not need to be exposed outside the cluster
- Allocation of IP addresses is easier. Automatic assignment to nodes is easier.
- No interference from traffic external to the cluster.

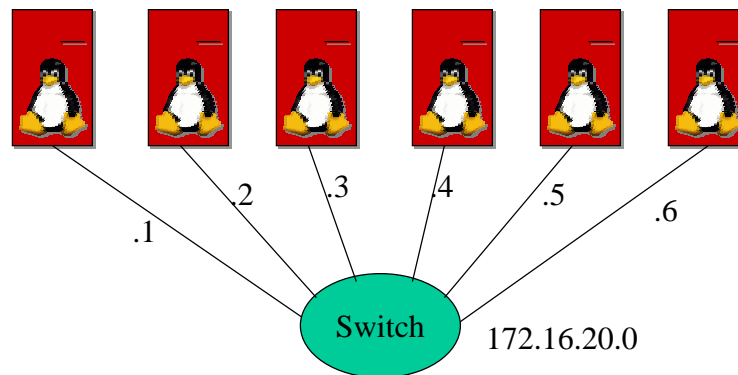


But...

- Users want connections to the outside world directly from cluster nodes.
- Users want high performance.



A Private Network



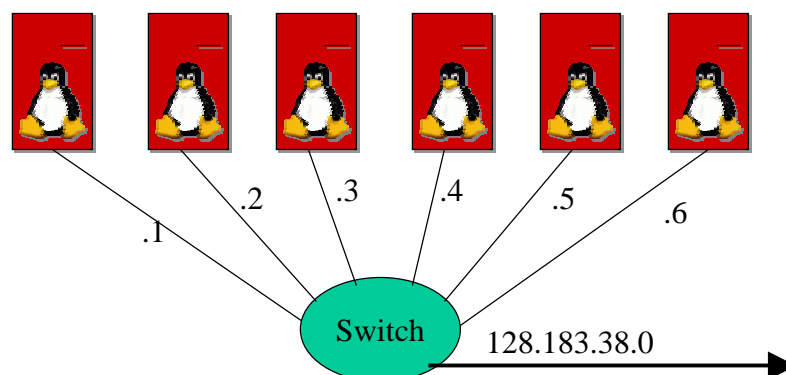
MCS

SC99 Tutorial Page 65

ACL

A Public Network

- All nodes are “on the internet”



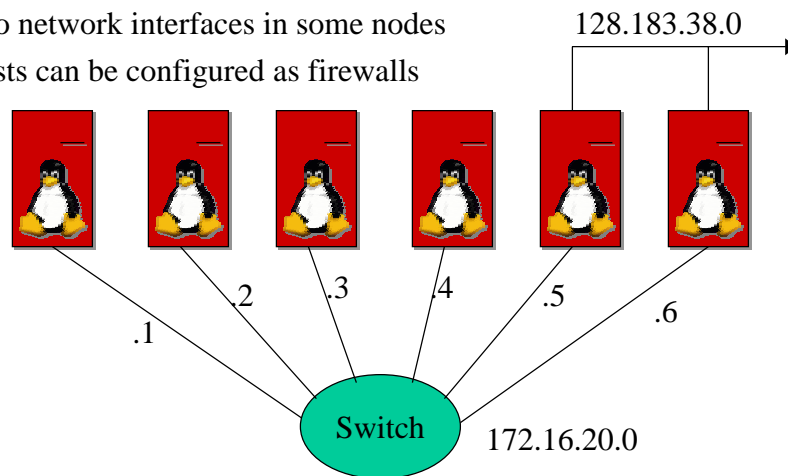
MCS

SC99 Tutorial Page 66

ACL

Gateway Nodes (+Masquerading?)

- two network interfaces in some nodes
- hosts can be configured as firewalls



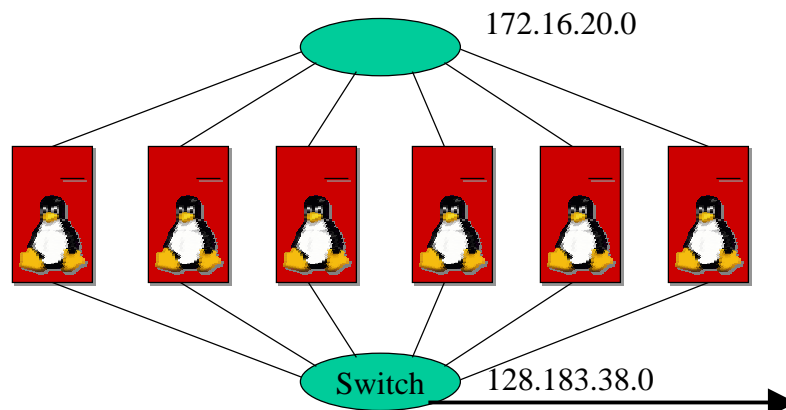
MCS

SC99 Tutorial Page 67

ACL

Two networks: 1 public, 1 private

- Routing/firewalling can be complex



MCS

SC99 Tutorial Page 68

ACL

IPC For Parallel Programs

- MPI
 - Industry standard. Your code will work everywhere.
 - Allows scalable programs
 - Designed to be integrated into production environments
- PVM
 - Not quite a standard
 - Performance limitations
 - Designed for “personal parallel computers” – difficult to integrate into a production system as it includes an “operating system”
- Virtual Shared Memory
 - No standards.
 - Researchy area.
 - No data on production environments.



SC99 Tutorial Page 69



Fundamental Choices

- Management Approach - Database, Hierarchical.
- Type & Number of CPUs
- Control Fabric - Have one.

- Number of Networks
 - We recommend switched fast ethernet + something better
- Type of High Speed Interconnect
 - Depends on \$ and requirements
- Public or Private Network
 - Depends on requirements



SC99 Tutorial Page 70



Chapter 5: Purchasing and Installation



Start with Solid Planning

Purchasing
and Installation

Write these Documents

- Overall System Diagram
- Life cycle & expectations
 - Machine lifetime (3 years?)
 - Delivered cycles (90% uptime?)
 - Funder's expectations...
- Purchase Timeline
- Installation Timeline
- Network Diagram
 - 1 diagram per network.
 - Show external connectivity.
- If you have them:
 - Serial Infrastructure.
 - Power Infrastructure.
- Racks Diagram
 - What goes in which rack where.
 - Cable management plan.
 - Floor plan for racks & cables.
- Power Planning
 - Which racks are wired on which circuit.
- Installation Guide
 - What goes in which rack where, in what order.
 - Cable location, labeling.
 - See MCS tools page for example.



SC99 Tutorial Page 72



Important Considerations (1)

- User profile and infrastructure (!compute_nodes)
 - Home directory disk space, NFS servers
 - Compile servers
 - Scratch disk, visualization, tape archives, etc
- What is the physical space?
 - Raised floor?
 - Air cooling units, and their ratings
 - AC Power conditioning
 - How tall can racks be? Will they fit in the doors?
 - How much AC power is available, how many circuits? Plug types?
 - Environmental Safety
 - etc...



SC99 Tutorial Page 73



Important Considerations (2)

- How much do you have to spend?
 - Software licensing
 - Hardware
 - People time
 - Administrators
 - Developers (new tools)
 - Software Engineers (help port applications, tune, hold hands)
- THEN contact the vendors and request non-binding information (save everyone's time)
- Based on what the vendors can deliver, iterate again on proposals and design documents.



SC99 Tutorial Page 74



Planning Checklist

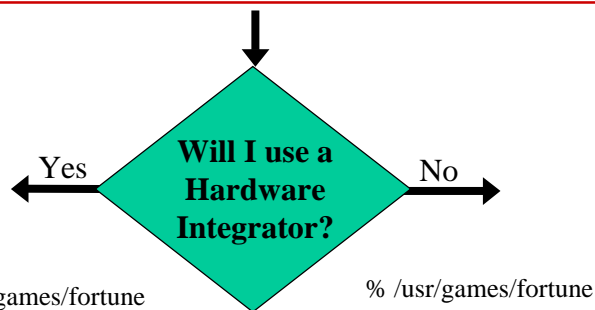
- Parts that have been on our purchase lists:
 - Computing & Login Systems
 - Processors, memory, disk, network, ..
 - Minimal video
 - Management Systems
 - Processors, memory, disk, network, ..
 - Other cards? Storage, more network, serial, ...
 - Other Systems
 - Visualization, storage, real-time, ...
 - Details vary widely.
- Network hardware
 - Fast & slow networks
 - Hubs/switches for cluster
 - Connectivity to world
- Serial Concentrators & Adapter Cards
- Power Boxes
 - Network for power boxes
- Cables, cables, cables
- Cable adapters
- Racks
 - Fans, doors, side panels



SC99 Tutorial Page 75



The First, Most Critical Choice:



Your Cluster will arrive pre-installed with Linux, after posing for pictures, you will spend your time on the hard part, software integration and administration.

Welcome to hell. Your gnashing of teeth will not fix your compatibility problems, and your new skills for attaching IDE cables is preparing you for a career at BestBuy. Remember, your time is free, and caffeine will help you get more done.



SC99 Tutorial Page 76



What Hardware Integrators can Provide

- A single, unified warranty. Failed nodes get sent to integrator.
- Integrated components, fewer surprises:
 - HW compatibility, physical form, heat, burn-in, etc.
- Preinstalled Linux, complete with appropriate drivers
- A small loaner system (or remote access) to test and benchmark the applications you really care about
- On-site installation
- Suggestions for system components
- Experience! (“we see power supplies die once a month on those systems”)
- Coffee mugs, T-shirts, and free lunches



SC99 Tutorial Page 77



There are Many Hardware Integrators

Just a Few:

- Compaq (www.compaq.com)
- Dell (www.dell.com)
- SGI (www.sgi.com)
- Penguin Computing (www.penguincomputing.com)
- Alta Tech (www.altatech.com)
- VA Linux (www.valinux.com)
- DCG (www.dcginc.com)
- Paralogic (www.plogic.com)
- Microway (www.microway.com)
- Hitech USA (www.hitech-use.com)



SC99 Tutorial Page 78



What About Software Integrators?

- There are companies now that will integrate EVERYTHING, and supply the site with complete support.
- From a recent press release:

Reston, VA – One of the fastest computer systems in the world has just been acquired by the Department of Commerce to help the National Oceanic and Atmospheric Administration (NOAA) further improve existing weather forecast models and develop new ones, Commerce Secretary William M. Daley announced. The \$15million contract has been awarded to High Performance Technologies, Inc. (HPTi) of Reston, Va., to provide a High Performance Computing System to NOAA's Forecast Systems Laboratory (FSL), located in Boulder, Colorado.

- Talk to Greg Lindahl (glindahl@hpti.com)



SC99 Tutorial Page 79



Talking to the HW Integrator

This is not a friendly place for the novice

- Remember that the integrator is assuming risk in exchange for money.
 - Integrating the warranties
 - Delivery of components in time for the cluster (e.g. memory price spike)
- Do your research, and know the prices for components.
- Only 1/10th of the detail demons:
 - Who installs what?
 - Acceptance tests
 - Who pays for return shipment of dead nodes during the warranty?
 - How will disputes be resolved?
 - Insist on uniformity (serviceability) where possible (Control node has same motherboard as compute nodes), etc.
 - Specify **manufacturer** part number for components to avoid confusion
 - Specify “automatic” rules for late delivery, etc



SC99 Tutorial Page 80



Shipping

- Will most likely arrive on one or more trucks. Make sure that works at your site.
 - Loading docks, carts, ...
- If possible, get phone # of driver to communicate drop-off location and key details.
- Keep very careful track of how many boxes arrived when, and if any were damaged.
- Have a plan for where boxes are to be stored before installation.



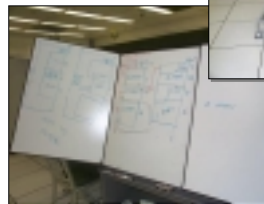
MCS

SC99 Tutorial Page 81

ACL

Installation

- Make sure you and your vendor understand who will do what.
 - If it's you, buy a power screwdriver.
- Expect 30-40% of the time to be on the first rack or two, working out the last details.
- Make sure the plan includes the ability to remove nodes for service.
- Have some type of early software verification test, so that you can test the installation while the vendor is still on-site.



MCS

SC99 Tutorial Page 82

ACL

Fundamental Choices

- Management Approach - Database, Hierarchical.
- Type & Number of CPUs
- Control Fabric - Have one.
- Number of Networks - At least two.
- Type of High Speed Interconnect
- Public or Private Network
- How Much You Build Yourself
 - Only as much as you have to.



SC99 Tutorial Page 83



Chapter 6: Cluster Administration



Chapter 6 - Cluster Administration

All those things that make the cluster work... but that the user only notices if they break:

- Physical Management
 - Power
 - Console
 - Bios
- Naming and Addresses
- Booting
- OS Management
 - OS Installation
 - OS Configuration
- File Synchronization
 - File Systems
- Applications
 - Installation
 - Environment Setup
- Monitoring
 - Installation
 - Steady State
 - Debugging
- Accounts
- Security
- User Notification

You need a strategy for each of these.



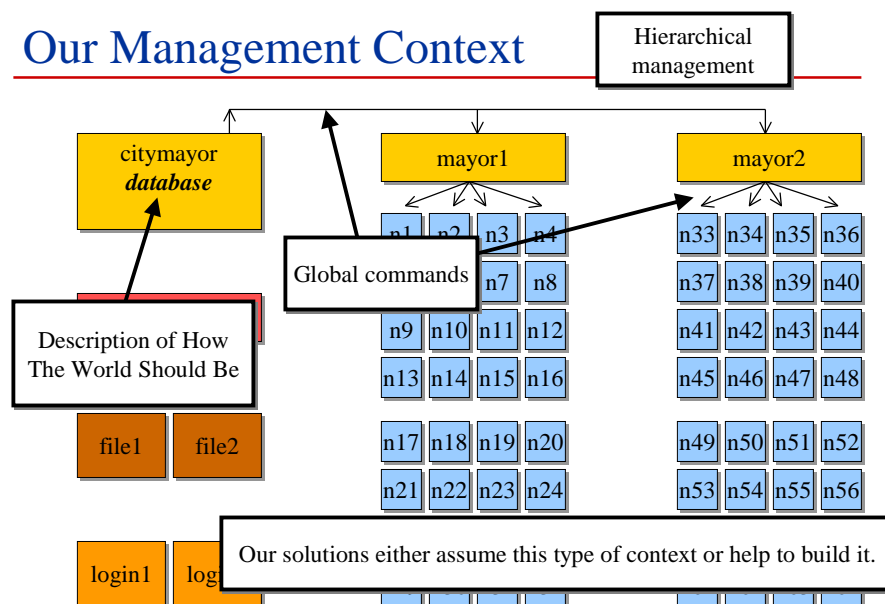
MCS

SC99 Tutorial Page 85



Our Management Context

Cluster Administration



MCS

SC99 Tutorial Page 86



Naming - Issues

- Hostnames are important... Choose wisely.
- Hostnames actually refer to network interfaces.
 - Users need to use the right hostname to get the right network performance.
- One host will often have multiple hostnames.



- Some questions to ask:
 - What route does a packet from n42 to n61 take?
 - How can you change that?
 - What does “n42” mean in this case?



Naming - Recommendations

- Pick something you can type. Bad: c001t003r01n23
- Pick something you can auto generate. Bad: sparky, fuzzy01, fuzz02, server ...
- Be consistent. Bad: node01, node2, ...
- Don't imbed too much information in names. Bad: compute23_i386_2G_9G
 - Names should be a hook to get that information from a database.
- Think about what the users need. Probably: node1 - node32
- Think about reverse name lookup.
- Think about what the routing issues will be.
- If possible, the generic node name should point to the fastest path in and out. I.e. node1 -> node-myr
- Add cnames for personalized names. “cluster” -> login1
- Document your naming convention.



Naming - MCS Solution

<clustername><use>[<number>][-<interface>]

Nodes:

ccn1 ... ccn256

Viz:

ccviz1 ... ccviz32

Login:

cclogin1 ... cclogin4

Network Specifics:

ccn1-feth: fast ethernet on node 1

ccn1-myr: myrinet on node 1

ccn1-hpn: "current" highperf net

ccn1 points to ccn1-hpn

Mayors:

cct1m ... cct8m

Power boxes:

cct1p1 ... cct1p5

Paper URL: <http://www.mcs.anl.gov/systems/papers/chi/>



mcs

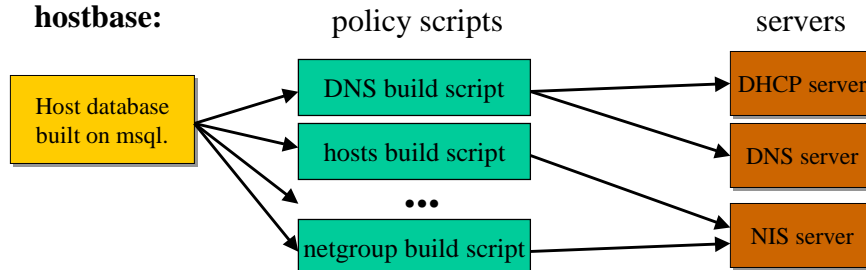
SC99 Tutorial Page 89



Naming - Tools

- Hostname implementation is generally tackled at the site level, not the cluster level. (But this approach will work in either case).
 - Private networks complicate this.
- We highly recommend the use of a database backend to drive all hostname-related stuff.
- An example is "hostbase", on the MCS systems tools web page:
 - <http://www.mcs.anl.gov/systems/tools/>

hostbase:



mcs

SC99 Tutorial Page 90



IP Address Ranges

- For public networks, use whatever address ranges are available to you at your site.
- For private networks, use any of these address ranges:
 - 10.0.0.0 - 10.255.255.255
 - 172.16.0.0 - 172.31.255.255
 - We recommend this one... Lots of address space, and no need to do your own subnetting.
 - 192.168.0.0 - 192.168.255.255
- Before this, you'll need to have thought about routing and such, which will affect your address choices.
 - Note - MCS likes switching... avoid routing if you can.



Booting

- Getting the first boot right is surprisingly tricky.
 - (So is deciding how to do it.)
- During the first boot process, in some order, you will:
 - Boot the first image, which may not be the last image.
 - Assign an IP address to the node.
 - Possibly get an image onto the node.
 - Possibly configure the OS on that node.



First Boot Mechanisms

- **Boot Floppy**
 - Enough of a kernel and image to get going.
 - Typical next step is to NFS mount larger system.
 - Options:
 - A different floppy per node (bad move).
 - One floppy for whole system.
- **Pre-loaded Hard Drive**
 - Good luck.
- **Boot CDROM**
 - Same as the floppy, except:
 - more \$ into hardware.
 - more space on CD.
 - Can use CD as filesystem instead of NFS.
 - more magic to get it to work.
- **Netboot Support in Ethernet Card**
 - Intel PXE, for example.
 - Requires DHCP & TFTP servers.
 - DHCP with PXE info in it.
 - Actual details vary widely dependent on your hardware.



SC99 Tutorial Page 93



Booting - Diskless or Local Disk

- | Diskless Booting | vs. | Local Disk |
|---|-----|---|
| <ul style="list-style-type: none"> • The OS lives on a server and is NFS exported to the nodes. • The nodes, if they have local disk, use it only for swap and temporary storage. • Booting happens via DHCP/BOOTP and TFTP. • Diskless booting is much easier to administer. <ul style="list-style-type: none"> • All management is done on the server, where the file systems of many nodes can be accessed simultaneously. | vs. | <ul style="list-style-type: none"> • A copy of the OS resides on the local disk. • You have to get the OS onto that disk and maintain it. • Local disk images are a bit more flexible. • Local disk will generally perform better, depending on network, server, and app characteristics. • Potentially more fault tolerant. |



SC99 Tutorial Page 94



Address Assignment - Issues

- Option 1 - Hardcode the address in the OS image
 - Tricky.
 - The OS installation has to differ across nodes (undesirable), or ...
 - You must use tools that figure out what address to use.
 - (And these often need net access.)
 - May be necessary for some network types.
 - If you do hardcode it, use configuration management (later topic) to handle it.
- Option 2 - Assign the address via a network service.
 - Several options....



Address Assignment - Network Services

- DHCP - Dynamic
 - Pro: Easy
 - Con: You don't know which physical node has which hostname, and it will vary.
- DHCP - Static
 - Pro: You can find your hosts by name.
 - Con: Initialization is quite tricky because of the need to map ethernet addresses to IP addresses in the DHCP table. Ways to get the ethernet address include:
 - Type them all in
 - Staged/time-delayed boot
 - Read them on first boot, then build the database from that.
 - SNMP query network boxes
 - Serial line reading
 - Consider what happens when you replace some hardware.
- BOOTP
 - Basically a subset of DHCP... Still has that table initialization problem



Address Assignment - Recommendations

- Your options depend a lot on your hardware.
 - Your hardware depends a lot on your budget.
- If at all possible, go with static DHCP.
 - Simplifies the OS image to use DHCP.
 - Simplifies debugging to have static hostnames.
- MCS solution (illustrated later)
 - The mayor knows which host is attached to which serial line.
 - During the first boot / install sequence:
 - the node displays its ethernet address to the mayor
 - the mayor updates the DHCP tables
 - The node then uses DHCP after that.



Subsequent Boot Mechanisms

- Your options are the same as the first boot, but:
 - You probably have DHCP and other servers configured.
 - You may have built your local disk during the first boot.

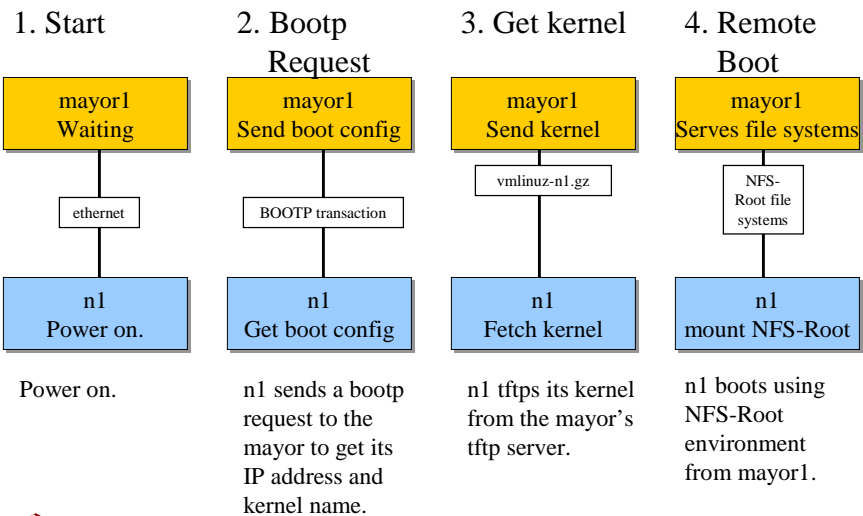


Booting Details

- In the following slides, we'll describe two different booting scenarios in detail:
 - Berkeley's Diskless Boot setup
 - MCS's Local Disk Boot and Build setup



Diskless Booting - TFTP/NFS Root



Booting - Setting up Diskless Booting

1. Create a network bootable kernel image.
 - Compile the primary network driver into the kernel.
 - Enable DHCP/BOOTP auto-configuration and NFS-Root support.
 - (See later slides about how to build an image.)
2. Create NFS-root tree for each node on server.
 - Most of the environment can be shared across nodes.
 - e.g. /usr, /lib, etc.
 - Other portions are individual to the nodes.
 - /dev -- must be private to a node for technical reasons
 - /etc -- some files may be shared, others are node specific
 - /tmp, /var
3. Export the nodes' root filesystems via NFS.



Booting - Setting up Diskless Booting

4. Set up DHCP and TFTP servers.
 - Configure with static IP address.
 - Recall previous discussion on how to get ethernet addresses...
 - Can use same kernel everywhere, but trivial to try a new one!
5. Configure the node to boot via BOOTP/DHCP and TFTP.
 - If you have BIOS support (alphas do), use that.
 - If you have ethernet card support (netboot), use that.
 - Otherwise, boot from floppy.

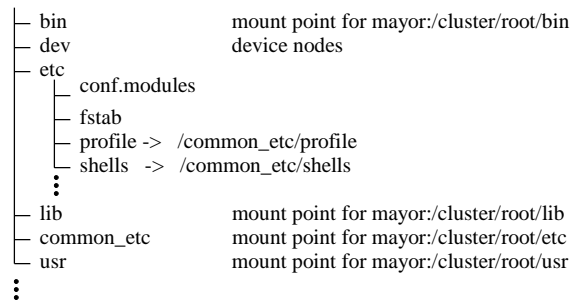
Notes:

- Kernel flags are important:
 - `<kernel_name> root=/dev/nfs rw nfsroot=<serverip>:/mnt/pnt ip=<node_ip>:<serv_ip>:<gateway>:<netmask>:<hostname>:<device> :none`
- Make sure the client kernel you boot has NFS server enabled for locking purposes.



NFS Root Structure (NERSC)

- Each node has a separate NFS root mount point.
 - Most directories in root fs are mount points for shared read-only filesystems.
 - /dev, /tmp and /var are private to a node.
 - /etc entries consist are either node specific, or are symbolic links to a shared cluster etc.



NFS Root -- Image Creation and Mgmt

- Creating a boot filesystem by hand is tedious.
- bccm** -- Berkeley Cluster Configuration Manager
 - Automatically creates NFS root tree for cluster nodes.
 - NFS root tree contains mount points for shared filesystems (/usr, /lib, /bin, /sbin, etc.), shadowed /etc, and full /dev
 - Configuration classes allow node customization.
 - Performs node- and cluster- specific substitutions within files.
 - e.g. the default NERSC cluster fstab has entries like:


```
@SERVERIP@:/cluster/lib /lib nfs ro 0 0
@SERVERIP@:/cluster/bin /bin nfs ro 0 0
```
- bccm also provides configuration management by forcing admins to strictly track all changes to base install.



Booting - MCS Approach

- Our requirements:
 - Flexible & dynamic OS installation on nodes.
 - Performance will occasionally be a critical issue.
 - Thus - we generally boot the OS on local disk but can also boot diskless.
- We have one boot floppy for the entire system.
- The floppy waits on commands from the mayor.
- The mayor references the cluster database and tells the image to:
 - Boot diskless from the mayor
 - Installs an image onto the local disk, or
 - Runs diskless from the mayor
 - Boot directly from the local hard drive

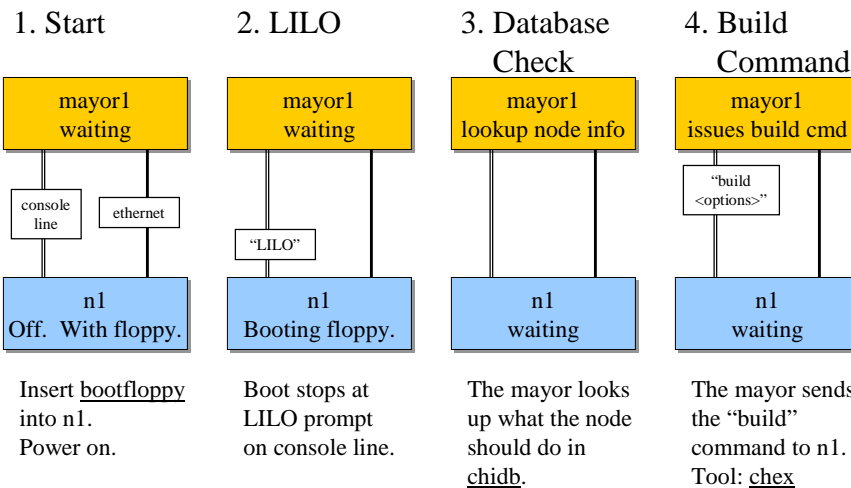


MCS

SC99 Tutorial Page 105



Booting - MCS Approach



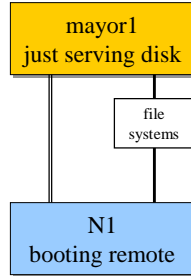
MCS

SC99 Tutorial Page 106



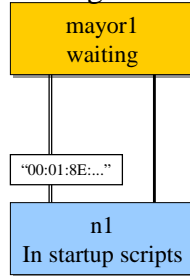
Booting - MCS Approach, Continued

5. Remote Boot



n1 boots using remote disk from mayor1.

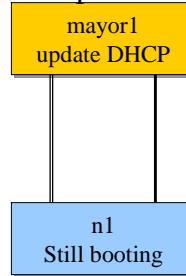
6. Ethernet Register



The boot process announces the ethernet address to the mayor.

Tool: build image

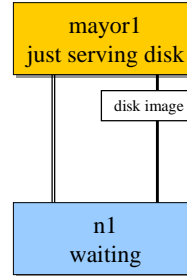
7. DHCP Update



The mayor updates the DHCP server with the ethernet address.

Tool: chex

8. Build Cmd



The boot process loads an OS image from the mayor onto disk.

Tool: diskimager



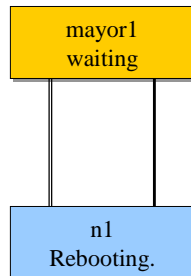
MCS

SC99 Tutorial Page 107



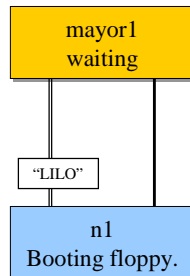
Booting - MCS Approach, Concluded

9. Reboot



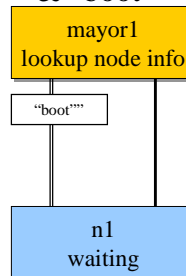
n1 reboots.
(From floppy.)

10. LILO.



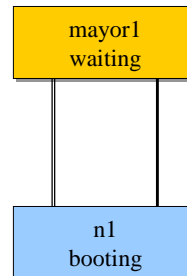
Boot stops at LILO prompt on console line.

11. Database & "boot"



The mayor looks up the node in chidb and says "boot".

12. Boot



N1 boots from the new image on local disk. The IP address comes from DHCP.



MCS

SC99 Tutorial Page 108



OS Image Management

- What is Image Management?
 - Getting the OS to the node.
 - Configuring it for the first use (“build”).
 - Configuring it over time.
- Which brings up some questions:
 - What is the OS?
 - Why does it need to be configured?
 - Why do we care?
 - What is the air speed velocity of an unladen swallow??
- It’s philosophy time.
 - Break out your coffee or beer and scoot up to the white board.
 - Bonus points for people with battle scars.



SC99 Tutorial Page 109



The Primary Issue is Change

- In an ideal world:
 - A new version of an OS...
 - ... would just work...
 - ... on all nodes, unchanged.
 - ... could, run, untouched, until the next OS release ...
 - ... or two.
 - You’d never need to...
 - ...add services.
 - ... install patches.
 - ... update network parameters.
 - ... debug problems.
 - ... install new apps.
 - ... wake up early.
- But in reality, depending on your environment and requirements:
 - The OS usually needs work.
 - Every node needs to be a little different.
 - You will have to make regular changes on your systems.
 - You’ll need to support many different applications and libraries.
 - You’ll have users who both want the same library, but want different versions of it.
 - Several people administer these systems at the same time.
- OS Image Management is really about change management.



SC99 Tutorial Page 110



Node Things, Categorized by Change

Things on a Node	Example	Issue
OS <div> <div>Base OS</div> <div>Base Mods</div> <div>Configs</div> </div>	Kernel, /, /root /usr, /bin, /lib, ...	Source: vendor Typically only changes with OS change.
	Replacements, links, permissions, ...	Source: admin One set of changes, made to base OS.
	/etc changes, crontabs, patches, ...	Source: admin Continual change for fixes, new features.
Applications	emacs, perl, mathematica, ...	Source: vendor & 3rd party Changes for new software, bugs, ...
User Space	Home directories, job input and output	Source: user Changes constantly



SC99 Tutorial Page 111



OS Image Management Strategy

1. Build the “base” system image.
The OS release.
Changes you have to make to it to get it to boot and be happy.
2. Have a reasonable way to make changes to the OS over time.
The “configuration” system.
3. Eventually build a new base system image.
Consider rolling in those changes.
(It helps if you know what those were...)



SC99 Tutorial Page 112



Building the OS Image

- The “everything” approach:
 - Install base Linux distribution onto a system by hand.
 - Install other software, RPMs, and whatever else.
 - Make any changes to the OS.
 - Store the results as the “image”.
 - Image type depends on how you plan to install it.
- Advantages:
 - Fairly easy.
- Disadvantages:
 - Not flexible.
 - Can be hard to debug weirdness.
- The “minimal” approach
 - Install base Linux distribution onto a system by hand.
 - Figure out how you want the final image to look.
 - Build a set of scripts and a list of RPMs that gets you to that image.
 - Store the result as an “image”, and invoke the scripts as part of the build process.
- Advantages:
 - More likely to work on different types of nodes.
 - Easier to upgrade over time.
- Disadvantages:
 - Hard to get just right.



SC99 Tutorial Page 113



Getting the OS Image Onto Disk

- Diskless
 - Build it on the server.
- Incremental OS Load
 - Boot mechanism + Build Script + Network repository
 - Kickstart + RPMs
 - Boot floppy + RPMs or NFS
 - CD + ...
 - This is the standard way to build a standalone Linux workstation.
- Physical Disk Image
 - Basically a ‘dd’ of a working disk.
 - Handy for “other” operating systems.
 - Fairly easy to snag.
 - Very, very disk size specific.
- Logical Disk Image
 - Partition information for disk.
 - Files to put into those partitions.
 - tar, dd, or copy of net disk
- diskimager handles both of the above.



SC99 Tutorial Page 114



Tools for Configuring the OS

- Cfengine - Mark Burgess
 - <http://www.iu.hioslo.no/cfengine/cfdetails.html>
 - Model:
 - Maintain a class-based description of each system.
 - Each system runs Cfengine and attempts to conform to description.
- cfg / sanity - MCS
 - cfg is a tool that keeps a database of configuration files, and knows how to install them into a system.
 - sanity is a framework for ensuring a node is at a known state.
- NERSC diskless tools
 - bccm, for managing the nodes' configuration files on the disk server itself.

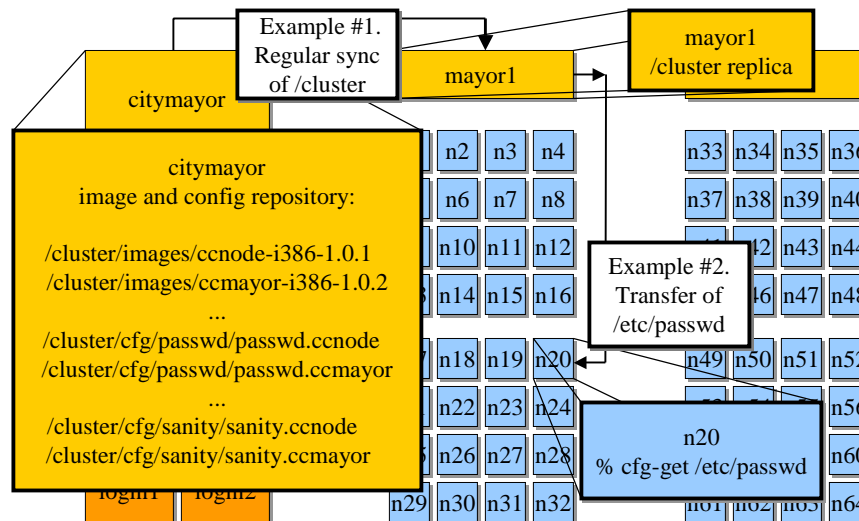


MCS

SC99 Tutorial Page 115



OS Image Management - MCS Solution

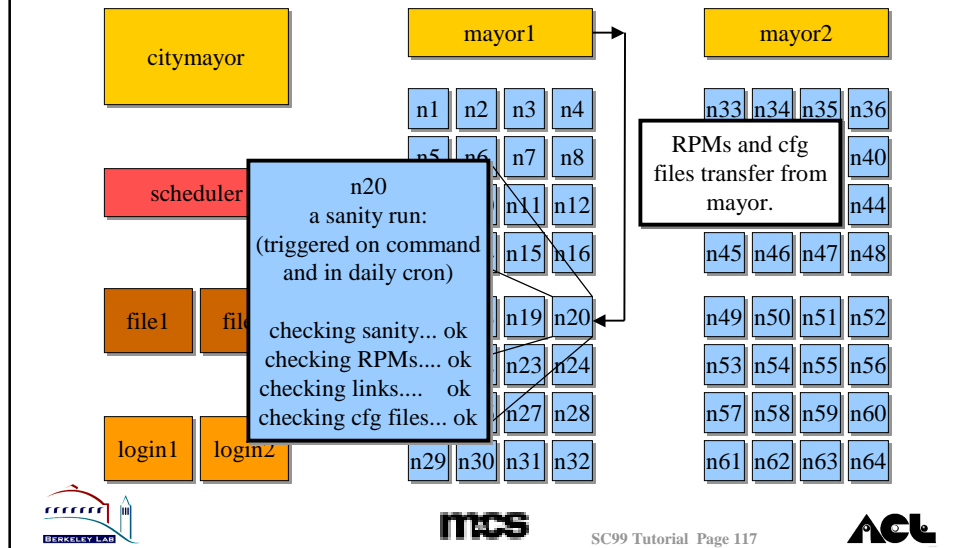


MCS

SC99 Tutorial Page 116



OS Image Management - MCS Solution



Global Commands

- Global commands are commands that are run from one spot and execute across the entire cluster.
- The trivial, stupid, but-mostly-works approach:
 - # foreach \$h (<list of hosts>); rsh \$ <command>; end
 - Serial, slow, unreliable.
 - But... many scripts are based around this model.
- There are quite a few prototype tools out there for this. We can't really recommend any of them yet. Scaling is a big issue.
 - prsh (beowulf tool)
 - Other tools on freshmeat.net and in development.
 - Research efforts:
 - ptools from Argonne
 - Multicast command distribution



MCS

SC99 Tutorial Page 118

ACL

File Systems

- File systems are perhaps the biggest problem Linux clusters face.
- Local Filesystems
 - Looks like ext2fs for now.
 - XFS from SGI announced, but not available.
- Global file systems: for a common name space across the cluster.
 - NFS
 - Doesn't scale well above a hundred nodes.
 - Watch out for parallel core dumps!
 - Set 'limit coredumpsizes 0'
 - AFS?
- Parallel File Systems
 - None in production.
 - Research efforts:
 - pvfs from Clemson University
 - others
- Definitely an important area for future research and development.



NFS Setup

- NFS setup and tuning can have a big effect on performance.
- Use knfsd for good performance
- Set number of knfsd threads at least equal to number of clients
- Version 1.5 of knfsd is currently unstable. Use 1.4.7 instead.
- Get knfsd from
<ftp://ftp.valinux.com/pub/support/hjl/knfsd/knfsd-1.4.7.tar.gz>
- For more information on NFS configuration/tuning, see
<http://pdsf.nersc.gov/talks/hep-nfs-fall-99/index.html>



Synchronization Tools

- In order to cope without a global, scalable file system, we synchronize key file systems.
 - Typically done nightly, from a main file system to many replicas.
 - Useful for mostly static data. Not good for dynamic (user) data.
- Options:
 - rsync
 - rdist
 - fmirror - based on ftp
 - See MCS pages for [synchronization scripts](#) for these tools.
- Under exploration:
 - multicast file copy



SC99 Tutorial Page 121



Application Installation - Issues

- Applications: all those thing you install after you've installed the OS. Sort of.
 - This strategy is closely related to OS Configuration and File Distribution.
- How much of an environment do you really want on the nodes?
 - Minimal: Just the Kernel, Ma'am.
 - Maximal: Workstation capability and then some.
 - Depends on what your users need.
- Where to install the software?
 - Local disk - better performance.
 - Remote disk - save disk space, consistency.



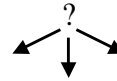
SC99 Tutorial Page 122



Application Installation - OS vs. Non OS

- Conceptual separation of the OS from added-on software will help save your sanity.
 - Upgrade applications independent of the OS.
 - Protect apps from OS upgrades.
 - Minimize OS configuration changes.
 - Keep the OS in /usr and /sbin.
 - Keep apps in /software.
- But, what, really, is the difference?
 - A free OS really confuses this.
 - Work very hard to keep them separate.

Classify:
gcc
glibc
sendmail



The OS	Freeware	3rd party
cat sed vmlinuz	perl emacs	Matlab Oracle



SC99 Tutorial Page 123



Application Installation - More Issues

- Installation Options
 - RPMs
 - Work well for local disk & single system.
 - Tend to walk all over the OS.
 - Use them if you can, they're very convenient.
 - Manual installation
 - You install software according to your policies:
 - Poured into /usr or /usr/local.
 - Carefully organized into specific directories.
 - ...
- How many versions of software do you keep around?



SC99 Tutorial Page 124



Application Installation - Recommendations

- Keep the OS separate from the applications.
- Install minimal software onto compute nodes.
 - RPMs are a good option for this.
- If you need a richer software environment, serve it from a file server.
 - RPMs are not a good option for this.
- Versions of Software
 - Keep around: new, current, old
 - (RPMs don't.)
 - Make sure users can get specific versions
 - Warn users of serious changes

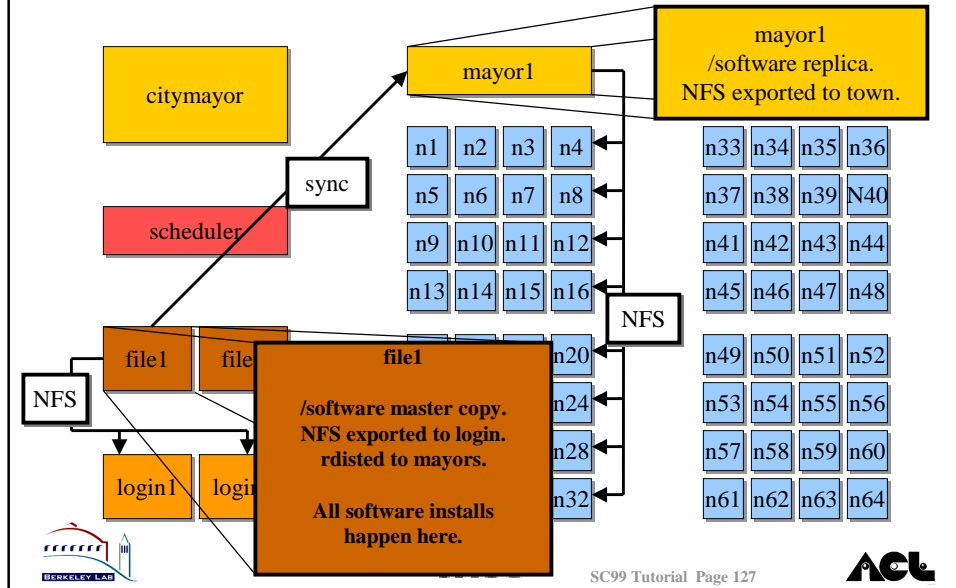


Application Installation - MCS Solution

- Nodes get minimal software. These are either:
 - Installed via RPM as part of the OS, or
 - Installed into /my (for local software needing performance).
 - These are managed by sanity.
- We have one master software server for the cluster.
 - Software is installed into /software on that server using pkg.
 - Each app lives in its own directory:
 - /software/linux/apps/packages/ssh-1.2.27
 - We have symlink “farms” for \$PATH:
 - /software/linux/apps/bin/ssh -> ../../packages/ssh-1.2.27/bin/ssh
 - This tree is mirrored to each mayor via disk synchronization.
 - Each node NFS mounts /software from their mayor.



Application Installation - MCS Solution



SC99 Tutorial Page 127



Application Environment - Issue

- For a user's environment to work correctly, certain environment variables must be set.
 - PATH, MANPATH, LD_LIBRARY_PATH, ...**
 - Some application specific variables.
 - Primarily an issue on any system where interactive (login) access will take place.
 - Also an issue for cronjobs, invoked jobs, and so on.
- Approaches to dealing with this vary:
 - do nothing
 - create nice startup dotfiles
 - consider having those source global startup files
 - inform users of what to do
 - set up a system (expected in a production environment)
- Tricky issues
 - Updating the environment over time.
 - Allowing users to switch versions of software dynamically.



MCS

SC99 Tutorial Page 128



Application Environment - Tools

- If you do nothing else, document correct settings:
 - in the default startup files
 - on a web page
- Options for systems to use:
 - MCS: softenv
 - Database of environment variables per package.
 - Users can pick and choose applications.
 - ACL: modules
 - perl-based rewrite of original Tk modules.
 - Nice support for dynamic environment changes.
 - Original modules
 - www.modules.org
 - Unclear if this is still under current development.



SC99 Tutorial Page 129



Accounts / Username Space

- You will need a way to distribute accounts across the cluster.
 - username <-> uid mappings
 - groups
 - home directories
 - passwords
 - ssh private keys
- Some of these are not so important on the computing nodes.
- The scheduler can/should handle some of this task.
 - In exclusive-mode use of the cluster, only one user has access to a node at a time. This is often implemented by modifying the passwd file.



SC99 Tutorial Page 130



Accounts - Tools

- NIS
 - Works on Linux, with some problems.
 - Scales to large, but not huge, size.
 - Possibly overkill for a cluster.
- Copying files around
 - Can result in password synchronization problems.
- NIS+
 - Not a lot of community buy-in...
 - Under active development for Linux.

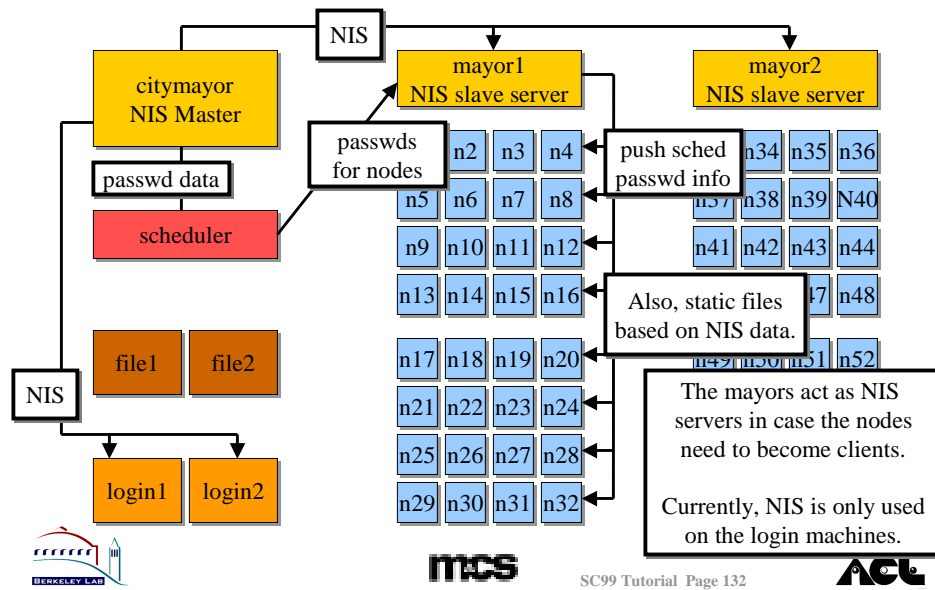


MCS

SC99 Tutorial Page 131



Accounts - MCS Solution



Security

- A Linux cluster on a public network is essentially a public network of Unix hosts.
- A Linux cluster on a private network is protected somewhat like a network of Unix hosts behind a firewall.
 - I.e. You still need to worry about security.
 - Particularly worry about security on the routing host.
- In either case, all standard Unix security recommendations apply:
 - Use tcp wrappers & ipchains.
 - Use ssh.
 - Turn off all services that you don't need.
 - Compute nodes probably need very few.
 - Regularly install patches.
 - Track bugtraq, security announcements, and key web sites.



Monitoring a Cluster

- A large cluster is an endless supply of raw data. What data is important, who wants it, and for what?
- Users:
 - How are things running? - performance monitoring
- Administrators:
 - Are things running correctly? - status monitoring
 - If not, what can be done? - debugging
- Monitoring is not a mature area yet.
 - There are lots of programs out there, doing many different things.
 - Unclear which scale, which don't.
 - No strong community agreement at this point.



Monitoring - System Administrators

- Installation / Control / Power / Ethernet
 - Lots of software, no clear winner.
 - Administrators use lots of standard Unix tools
 - Perl/Expect + [rsh, traceroute, ping, telnet to router, etc]
- Collecting historical log files (/var/log/*)
- Instantaneous status
 - Direct hardware monitoring: Power, fan, temp
 - Holy Grail: fault prediction
 - Anomalies (crash/failure, security, lethargy, etc.)
 - General status and performance



Monitoring - Users

- Performance monitoring for their programs (why is it so slow?)
 - TAU <http://www.cs.uoregon.edu/research/paracomp/proj/tau/>
 - VAMPIR <http://www.pallas.com/pages/vampir.htm>
 - MPE/Jumpshot <ftp://ftp.mcs.anl.gov/pub/mpi>
- Insight into machine behavior while their code runs
 - Supermon/Superview <http://www.acl.lanl.gov/software>



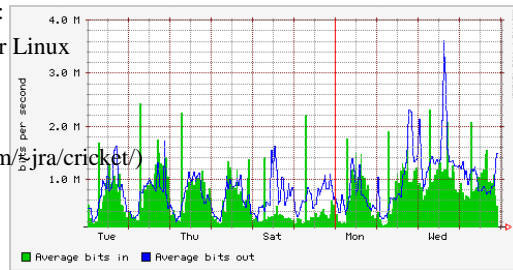
Collecting Log Files

Security and Accounting

Cluster
Administration

Monitoring

- Hierarchical collection is required for large clusters
 - Mayor collects logging data from compute nodes
 - Mayor data can then be collected into single location
 - Currently merging or collecting log files from the Mayors is ad hoc
 - Simple Perl: cron job, rcp files from Mayor, rollover old data..
- Data mining is generally site-based
 - detecting “su”, access denied, /var/log/pacct
 - Vaporware announcement:
 - Project accounting for Linux
- Syslogd
- Cricket (formerly MRTG)
 - (<http://www.munitions.com/~jra/cricket/>)



MCS

SC99 Tutorial Page 137

ACL

Instantaneous Monitoring

Automated Problem Detection

Cluster
Administration

Monitoring

- EMP port monitoring of hardware status via serial line
 - <ftp://ftp.valinux.com/pub/software/vacm/>
 - Integrated, hierarchical, gui, not yet ready
- rstatd/ruptime
 - Very primitive, limited information

```
%ruptime
node1      down 18+04:16
node2      up 16+14:40,  0 users, load 1.77, 1.72, 1.04
node3      up 11+03:14,  0 users, load 0.17, 0.16, 0.83
```
- rsh <node> <my-monitoring-program>
- Supermon / Superview



MCS

SC99 Tutorial Page 138

ACL

Supermon and Superview

Monitoring Clusters in Real Time

- Ron Minnich, Karen Reid, Matt Sottile
- You can see the software demonstrated at the LANL booth
- Goal: Fast, accurate, monitoring of **true** node performance
- Design:
 - Monitor hundreds of nodes at rates up to 100 Hz
 - Monitor at 10Hz without significant impact on the application
 - Monitor hardware performance counters
 - Collect a wide range of kernel information (disk blocks, memory, interrupts, etc)
 - All data is time-stamped, so events can be correlated



SC99 Tutorial Page 139



Supermon Architecture

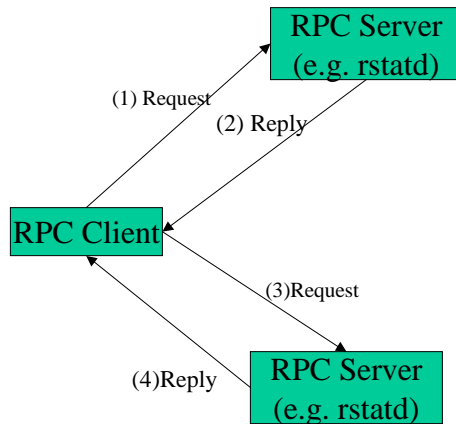
- Compute Nodes:
 - A modified rstatd accepts “get status” request from monitoring node
 - A modified Linux kernel adds a sysctl to get important system parameters without going through (slow) /proc.
- Monitoring Node (running Supermon):
 - Supermon uses vector RPC to stat Compute Nodes
 - Supermon concatenates data into a single stream
 - Supermon is a server, and monitoring programs can attach and slurp down the aggregated status information.
- Clients (such as Superview)
 - Connect to Supermon, get data as simple ASCII data stream
 - Supermon can filter data stream, and only send what is requested
- Hierarchical “mux” under development



SC99 Tutorial Page 140



RPC the Old-fashioned Way



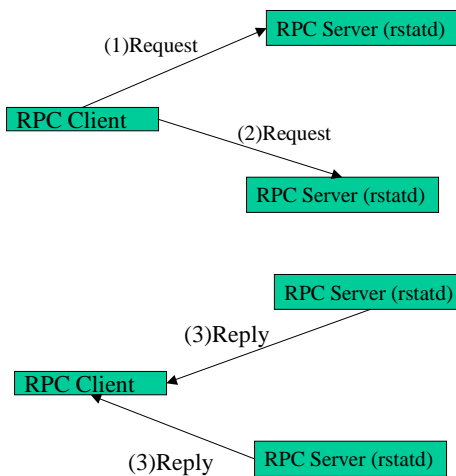
- Requests are sent to one host at a time
- The client waits for each reply before proceeding to the next server
- On a 100-node cluster this query would take 2 seconds



SC99 Tutorial Page 141



Vector RPC



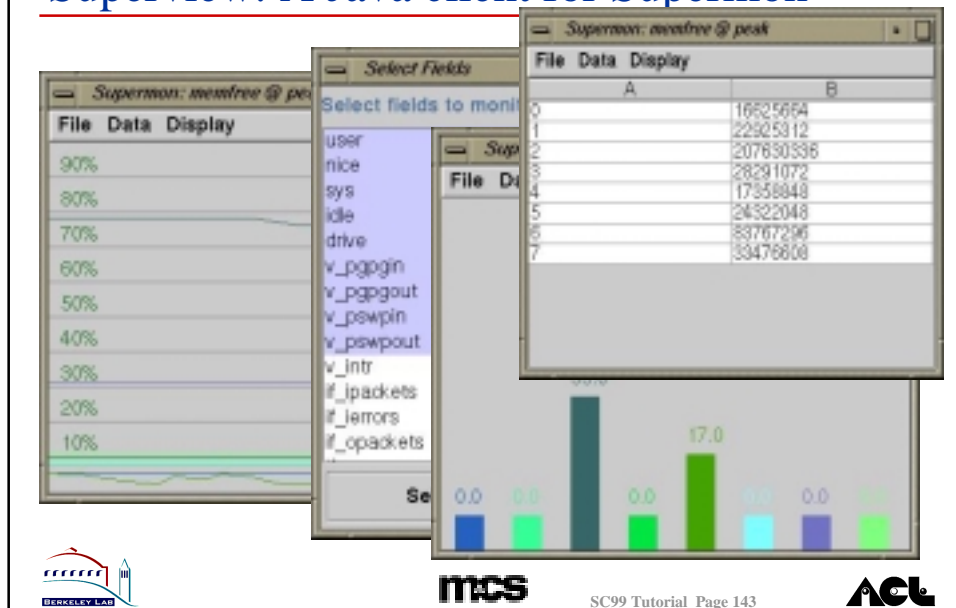
- Requests are sent out in groups
- Requests are processed at remote nodes in parallel, and replies come back “all at once”
- Once all requests are sent, replies are processed
- Requests for “missing” replies are resent as needed
- We can query a 100-node cluster in 20 milliseconds
 - This is 100 times faster than the non-vector RPC method
- Vector RPC performance makes cluster monitoring practical



SC99 Tutorial Page 142



Superview: A Java client for Supermon



SC99 Tutorial Page 143

ACL

Fundamental Choices

- Management Approach - Database, Hierarchical.
- Type & Number of CPUs
- Control Fabric - Have one.
- Number of Networks - At least two.
- Type of High Speed Interconnect
- Public or Private Network
- How Much You Build It Yourself
- Remote or Local Disk
 - Remote, unless you prefer otherwise.
- How You Manage Change



MCS

SC99 Tutorial Page 144

ACL

Chapter 7: Application Environment



Networks

IPC For Parallel Programs

- MPI
 - Industry standard. Your code will work everywhere.
 - Allows scalable programs
 - Designed to be integrated into production environments
- PVM
 - Not quite a standard
 - Performance limitations
 - Designed for “personal parallel computers” – difficult to integrate into a production system as it includes an “operating system”
- Virtual Shared Memory
 - No standards.
 - Researchy area.
 - No data on production environments.



SC99 Tutorial Page 146



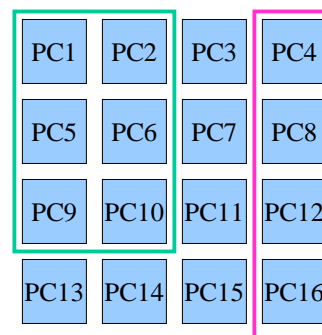
MPI Implementations

- MPICH from ANL/MSU
 - <http://www.mcs.anl.gov/pub/mpi>
 - Different transport layers available
- LAM from Notre Dame
 - <http://www.mpi.nd.edu/lam>
 - Designed for personal supercomputers. Some problems integrating a batch system and with an external process manager.
 - Faster and easier to use than MPICH for TCP-based communication and no batch integration.
 - Good support for code development.



Space Sharing (Revisited)

- Tightly coupled parallel programs must have dedicated compute resources.
 - Static load balancing in most parallel apps means loadave of 2 on one node of N reduces performance by at least 50%, not $O(1/N)$.
 - Synchronization delays reduce performance further.
- Standard solution: space-sharing.



MPICH Device Support

- **ch_p4**: the “TCP” device
 - Process startup and communication are intertwined. No modularity.
 - Performance is so-so (e.g. 120us latency)
- **via**: runs anywhere VIA does.
 - Process startup and communication are separate
 - High performance
 - Experimental. Available from BLD web site.
- **ch_gm**: Myrinet-specific.
 - Process startup and communication are intertwined.
 - High performance
 - Available from <http://www.myri.com>.
 - Installation is tricky.



Building MPICH for ch_p4

- Make sure to apply all patches. The base distribution is not updated.
- Configure tool makes life easier.
 - Specify `-arch=LINUX -device=ch_p4` (shared memory device not recommended)
 - Specify location of Fortran (`-fc=`), C (`-cc=`), F90 (`-f90=`) and C++ (`-c++=`) compilers explicitly. Otherwise you rely on correct configuration of PATH at build and run time.
 - Specify `--prefix=/usr/local/pkg/mpich-1.1.2`
 - If using ssh instead of rsh use `-rsh=ssh`



Running MPICH for ch_p4

- All hosts in cluster need to trust “mpirun” node without a password.
- User path should be set to bin dir of install directory.
- User-visible commands
 - mpicc/mpif77/mpif90 to compile and link (no options required)
 - mpirun -np N a.out



Things to Know About ch_p4 (1)

- Does not do space sharing.
 - Can only be fixed by a scheduler with “global” knowledge. We will show how to do this with PBS.
 - Decides where to run processes based on a hostfile in an obscure format
- By default starts up the first process on the “mpirun” node.
 - Not the right thing if you run from an interactive node.
- Orphaned processes
 - “kill -9” to the wrong process can leave others stranded, spinning
 - Something has to kill off those processes automatically (PBS prologue/epilogue)



Things You Should Know About ch_p4(2)

- Slow startup
 - Based on rsh/ssh
 - Approx 1 task/sec
 - There is not yet a general purpose workaround for this.
- TCP disconnects under severe congestion. No known workaround.
- Relies on command line args in process startup. Size of command line grows with number of nodes. Large clusters need workaround.



MVICH

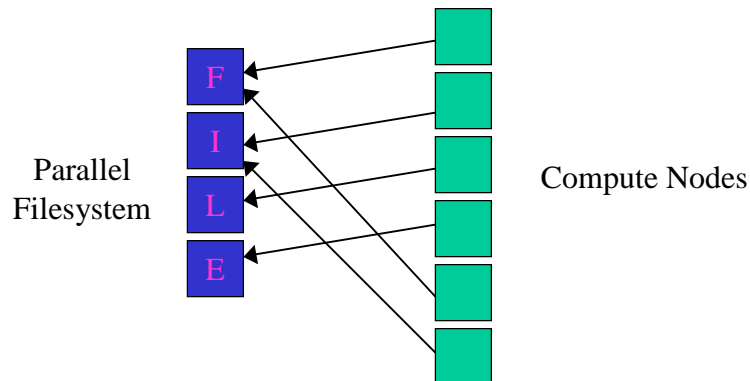
- MVICH is a VIA device for MPICH
- In “alpha” state. Available from BLD site.
- Requires a VIA implementation
- Provides very high performance on a wide range of networks.
E.g. 12us on Gigaset.
- Process management is modular.



Parallel I/O

Application
Environment

Parallel I/O



mcs

SC99 Tutorial Page 155

ACL

MPI-IO and ROMIO

Application
Environment

Parallel I/O

- Doing efficient parallel I/O is tricky.
- Need an efficient software interface *and* a parallel/scalable filesystem.
- MPI-2 I/O is the standard interface for cooperative parallel I/O
- The best/only implementation for clusters is ROMIO.
 - <http://www.mcs.anl.gov/romio>
 - Unfortunately there is not yet production level parallel filesystem to run it on top of!
 - Best bets are
 - NFS
 - PVFS: <http://ece.clemson.edu/parl/pvfs/>



mcs

SC99 Tutorial Page 156

ACL

Compilers

- Expect to buy one.
- C
- C++
- Fortran/90
- OpenMP (C and Fortran)



Compilers/x86

- C: gcc is fine
- C++
 - Newest gcc/egcs is good.
 - KCC (from Kuck and Associates -- www.kai.com -- has been necessary in some cases. Includes OpenMP support.
- Fortran
 - Portland group is the most widely used. Includes OpenMP support.
 - Other available from: Absoft, Fujitsu, NAG
 - g77 only as a last resort. (slower, limited syntax).



Compilers/Alpha

- C
 - gcc is ok.
 - Compaq C compiler is in beta release. Faster code.
- C++
 - g++/egcs is the only option
 - Compaq compiler being ported.
- Fortran
 - Compaq Fortran compiler is in beta. Much faster code.
 - Don't even think about g77.

<http://www.unix.digital.com/linux/software.htm>



Basic Math libraries

- Optimized BLAS for x86 available
 - <http://www.cs.utk.edu/~ghenry/distrib/>
- FFTs available in FFTW package
 - <http://www.fftw.org>
- Compaq math libraries for alpha
 - cpml: a drop-in replacement for libm.a
 - <http://www.unix.digital.com/linux/cpml/index.html>



Debuggers

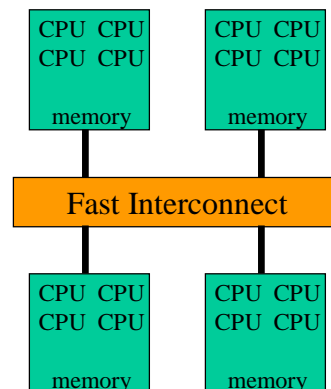
- Slim pickings here.
- The best parallel debugger, but also very expensive, is Totalview.
 - <http://www.etnus.com/products/totalview/index.html>
- See also the Data Display Debugger. It's not parallel, but it does have a nice interface.
 - <http://www.cs.tu-bs.de/softech/ddd/>



Programming Models for SMP Clusters

Warning... Religious Topic Ahead

- The Simplest Option:
- Ignore the architecture: **MPI Everywhere**
 - Advantages:
 - Very simple, very portable
 - Relatively easy to debug
 - Disadvantages:
 - Almost always extra memory copies on an already limited memory bandwidth node
 - Can have performance problems on cache-coherent shared memory architectures
 - Ignores the hardware support for shared memory



Threads & MPI

- Methodology:
 - 1) Begin with a standard MPI program, add thread parallelism by hand to sections of the 'node code'. [David Bader (dbader@ece.unm.edu)]
 - 2) Redesign the code for two levels of parallelism, and then decompose across nodes for msg passing, and threads for shared memory
- Advantages:
 - Pthreads are fairly ubiquitous. The single box program does not require msg passing, and is tuned for SMP and cache effects
 - Dynamic parallelism and load balancing is easier in shared memory
 - Extra data copies can be avoided
- Disadvantages
 - Thread safe MPIs are rare
 - Code is much more complex, and required lots of work, hard to debug



OpenMP & MPI

- Very similar in nature to Threads & MPI, but uses program directives to inform compiler about possible parallelism
- Methodology:
 - Begin with a standard MPI program, add OpenMP directives to the "node code" loops, and ask the compiler to parallelism and synchronize them.
- Advantages:
 - Compiler vendors are strongly supporting OpenMP
 - In depth understanding of threads and thread models is not required
 - Compilers can help
 - Extra data copies could be removed
- Disadvantages:
 - Code is much more complex, and required lots of work, hard to debug
 - Mixing compiler directives and semi-explicit parallelism is messy
 - MPI Thread safety can be a problem



Use a Programming Framework

- Principle: The source code of the application should not become more complex to support more sophisticated run-time layers
- Let the underlying run-time support for the framework work out all the complexities of executing code for the machine.
- Examples:
 - POOMA/SMARTS (www.acl.lanl.gov/smarts)
 - OVERTURE
- Fundamental scientific programming abstractions are presented to and manipulated by user. Direct message passing is avoided.
- The framework moves data (message passing) and understands parallelism, and can use threads for SMP parallelism.



SMP Programming Summary:

- Easy, simple, portable: MPI Everywhere
 - Unnecessary data movement and copies likely
- MPI + Threads
 - Requires more complex source code, but can deliver good SMP performance. Thread safety can be an issue
- MPI + OpenMP
 - Requires more complex source code, but can deliver good SMP performance. Special compiler required, mixing directives and explicit parallelism. Thread safety can be an issue.
- Parallel Programming Framework
 - Source code uses parallel programming abstractions, and is not made more complex for SMPs. RTS can improve execution behind the scenes. Requires a large investment in the framework and use.



Fundamental Choices

- Management Approach - Database, Hierarchical.
- Type & Number of CPUs
- Control Fabric - Have one.
- Number of Networks - At least two.
- Type of High Speed Interconnect
- Public or Private Network
- How Much You Build It Yourself
- Remote or Local Disk - Lean towards remote.
- How You Manage Change
- Use MPI. (No choice.)



SC99 Tutorial Page 167



Chapter 8: Process and Resource Management



Process and Resource Management

- How do we start jobs?
- Who decides where they are run?
- Who keeps users from getting in each other's way?



SC99 Tutorial Page 169



Running Jobs

For a production system, we want

- MPI jobs must be space-shared
- Utilization important: run jobs through the night; pack jobs.
- Complex scheduling policies
 - Fast turnaround for small jobs
 - Large jobs don't get starved
- "mpirun -np 4 a.out" should work!
- Partitioning system into batch/interactive wastes resources.
- All of this integrated with accounting



SC99 Tutorial Page 170



Process/Resource
Mgmt

Process anager vs. Resource Manager

- A **process manager**
 - Starts the processes in an MPI application on some nodes
 - Directs output/error to the terminal
 - Propagates signals
 - Knows what is running on the system
- A **resource manager**
 - Decides what jobs run on what nodes at what time.
 - Makes policy decisions and asks process manager to implement
 - Note: P4 contains a process manager, not a resource manager.
 - Note: Sometimes a resource manager is a host file



SC99 Tutorial Page 171



Portable Batch System

There are few systems that can do this.

Community is standardizing on PBS

- Open Source
- Modular: policy module (scheduler) is separate from batch system proper.
- Portable (many operating systems, processors)
- Fairly robust (but not perfect!)
- Can handle both batch and “mpirun a.out” jobs
- PBS is a distant descendent of NQS



SC99 Tutorial Page 172



Installing PBS

- Download from <http://pbs.mrj.com>
- Configure/build instructions at BLD site
 - Client and server installation is different
 - Installation is fragile
 - Configuration is tricky



PBS Servers

3 servers do the work

- pbs_server
 - maintains database of running/submitted jobs
 - accepts requests (submission, kill, query)
- pbs_sched
 - makes policy decisions. Queried directly by pbs_server
 - modular -- can be replaced with arbitrary scheduler
- pbs_mom
 - one runs on each compute node
 - starts user jobs
- nothing: runs on login/interactive node



External Schedulers: Maui

- The default scheduler (FIFO) is not sophisticated
- The “Maui Scheduler” was written for MHPCC, running on IBM SPs and using IBM’s Load Leveler for job management.
- Full bells-and-whistles 3rd party scheduler:
 - Excellent backfill algorithm
 - Queue control (priorities per project, etc)
 - Usage reporting
 - ...
- Can be integrated as a drop-in scheduler replacement in PBS.
 - First in operation on Linux in late October at the University of New Mexico.
 - Release date not yet specified at time of slide writing.



SC99 Tutorial Page 175



Queues in PBS

- PBS supports multiple “queues”.
 - Old-fashioned and generally unnecessary. Historically have been used to classify different sizes of jobs for dumb scheduling algorithm.
 - We recommend one queue
 - Rely on a sophisticated scheduler to sort out what/when to run.
- Remaining use of queues
 - Users can request different scheduling treatment.
 - Note: user specifies queue, not system.



SC99 Tutorial Page 176



Running a PBS job

- User requests number and type of nodes, and walltime limit.
 - `qsub -l nodes=4 -l walltime=5:00:00 job.script`
- PBS job is a shell script (including interactive jobs)
- PBS allocates N nodes to job, starts script on first node.
- Parallel application is started by “mpirun” inside script.



PBS Prologue/Epilogue

- Clean orphaned jobs
- Restrict/restore user accounts
- Scripts on BLD site



Problems with PBS

- Scalability -- one mom per compute node; linear algs.
- Doesn't understand SMP nodes without help
- Gets confused/hung when nodes go down
- Easily overloaded by too many back-to-back requests
- Configuration difficult, finicky, and not obvious
- Scavenger jobs not easily supported.
- Security: requires complete rsh-trust among nodes in cluster (Future: replaceable authentication module)
- Not cleanly separated from process manager in all cases.



Integrating MPICH with PBS

- Communication between PBS and "mpirun" is through a file containing a list of nodes. (\$PBS_NODEFILE).
- mpirun must be modified to know about \$PBS_NODEFILE and start machines on those nodes
- Modifications to mpirun (ch_p4 and ch_gm) are on BLD site. (Some mods available on PBS site are less robust)
- Node name translation is needed to run on non-default network
- Current solutions invoke rsh from mpirun. This means PBS does not know about the individual processes on other nodes.



Configuring PBS for SMP nodes

- Next version of PBS will support virtual nodes
- With current version, need to fool the software.
 - Space sharing by node, as usual
 - Process launch software (mpirun) must understand SMP nodes
 - Create machine file from PBS_NODEFILE by duplicating entries in order to start more than one process on a node.
 - For ch_p4, do not simply double process count because allocation of processes to nodes will tend to maximize inter-node communication.



SC99 Tutorial Page 181



Configuring PBS for Heterogeneous Nodes

- PBS node description file in \$PBS_ROOT/server_priv/nodes
- Add a set of attributes to each node.
- Nodes with fewer attributes are allocated first.

babel1	mem128	mem512	gige	myrinet
babel2	mem128	mem512	gige	
babel3	mem128	gige	myrinet	
babel4	mem128			



SC99 Tutorial Page 182



“mpirun -np 4 a.out” From the Terminal

- Users don’t like to write scripts, submit, wait for job to come finish. They want “mpirun -np 4 a.out” to work.
- Administrators can’t let users use mpirun outside of PBS because it won’t space-share and because system partitioning lowers utilization.
- PBS supports interactive jobs, but not in a way that easily allows single-command job execution.
- Solution: magic “pbsrun” script that cleverly wraps PBS commands to provide transparent single-command execution. Available at BLD site.



SC99 Tutorial Page 183



The Last of the Fundamental Choices

- Management Approach - Database, Hierarchical.
- Type & Number of CPUs
- Control Fabric - Have one.
- Number of Networks - At least two.
- Type of High Speed Interconnect
- Public or Private Network
- How Much You Build It Yourself
- Remote or Local Disk - Lean towards remote.
- How You Manage Change
- Use MPI - No choice.
- Your Scheduling Policy - PBS, because that’s all there is.



SC99 Tutorial Page 184



Conclusion

- We've described many of the issues and problems, our approaches, and some of the possible tools.
 - See Appendix A for pointers to tools.
 - See Appendix B for other references.
- The world of Linux clusters is maturing rapidly. This will be fun.



SC99 Tutorial Page 185



Appendices



Appendix A - Tools Mentioned

This appendix is a list of all of the tools mentioned throughout this tutorial. Many of them were written at one of our laboratories as part of this or related projects. Many others are from the open source community. Where possible, we have URLs for the software.

You may find updated or new tools at any of the main project pages at our sites.

For reference, here are those URLs:

- NERSC at Lawrence Berkeley National Laboratory:
 - <http://www.nersc.gov/research/bld>
- MCS at Argonne National Laboratory:
 - <http://www.mcs.anl.gov/systems/software/>
- The ACL at Los Alamos National Laboratory:
 - <http://www.acl.lanl.gov/software>



SC99 Tutorial Page 187



Tools - Booting and Images

Chiba boot floppy

- MCS - <http://www.mcs.anl.gov/systems/software/>
- A disk image of the floppy that used to boot Chiba City nodes. Includes a flexible kernel and a LILO that is configured to wait for the mayor to manage it over a serial line.

Chiba node RH 6.1 image

- MCS - <http://www.mcs.anl.gov/systems/software/>
- Our Red Hat 6.1 based image that we load onto the computing nodes.

Chiba build scripts

- MCS - <http://www.mcs.anl.gov/systems/software/>
- The scripts used to install the Chiba image onto a node.



SC99 Tutorial Page 188



Tools - Booting and Images

Chiba Boot CD

- MCS - <http://www.mcs.anl.gov/systems/software/>
- An image and instructions for making a boot CD for the IBM 4000R nodes.

Disk Imager

- MCS - <http://www.mcs.anl.gov/systems/software/>
- A tool that can create a disk image of a node and archive it, storing the disk as either a binary image or as file system contents. It can also be used to load the image of a disk onto a system.



SC99 Tutorial Page 189



Tools - Configuration

bccm - Berkeley Cluster Configuration Manager

- <http://www.nersc.gov/research/bld>
- A tool for managing the images and configurations of diskless nodes.

Cfengine

- Mark Burgess - <http://www.iu.hioslo.no/cfengine/cfdetails.html>
- Uses a high-level language to describe node configuration and makes modifications to systems based on the rules. Widely used across the systems administration community. See the end of this appendix for an example rulefile.



SC99 Tutorial Page 190



Tools - Configuration, 2

cfg

- MCS - <http://www.mcs.anl.gov/systems/software/>
- A configuration file management system. Cfg stores files in a central repository and installs them on individual systems.

sanity

- MCS - <http://www.mcs.anl.gov/systems/software/>
- A system that looks after the configuration of a particular node. It essentially runs a series of modules until they succeed, at which point the system is presuable “sane”. The modules can update configuration files, chek and load RPMs, change kernels, or whatever is necessary.



SC99 Tutorial Page 191



Tools - Serial Line & Power Management

chex

- MCS - <http://www.mcs.anl.gov/systems/software/>
- chex watches the console of a node, looking for particular expressions and sending messages.

conserver

- Open Source, originally from Purdue, modified elsewhere
- On the MCS web pages, but not written by MCS.
- Runs as a daemon on a server,, allowing network access to console/serial lines through a terminal server or serial concentrator.

chi_power

- MCS - <http://www.mcs.anl.gov/systems/software/>
- Allows one to control the power to any node or set of nodes in the cluster. Requires chidb and Baytech power control units.



SC99 Tutorial Page 192



Tools - Application Installation

pkg

- MCS - <http://www.mcs.anl.gov/systems/software/>
- A suite of tools to help manage large software installations. Handles multiple versions of software, symlink farms, and creation of web pages with documentation. As much a philosophy as a tool suite.

softenv

- MCS - <http://www.mcs.anl.gov/systems/software/>
- Tools to help build a user's shell environment based on instructions in one of their dotfiles. Works well with pkg.

modules

- Original TK version: www.modules.org
- ACL Update - <http://www.acl.lanl.gov/software/>
- Perl scripts to implement 'module' command for linux, making it easy to switch between compilers, message passing layers, and libraries



SC99 Tutorial Page 193



Tools - Databases

hostbase

- MCS - <http://www.mcs.anl.gov/systems/software/>
- An example of using a SQL database (mysql in this case) to keep all host-related information in one spot, and to use it to build DNS, NIS, and DHCP servers.

chidb

- MCS - <http://www.mcs.anl.gov/systems/software/>
- A SQL database of cluster information, including what OS image should be on what node, what nodes are on what towns, and so on. Used by several Chiba City tools.



SC99 Tutorial Page 194



Tools - Methods

ssh key management

- MCS - <http://www.mcs.anl.gov/systems/software/>
- Instructions for using ssh to run global commands securely across a cluster.

File synchronization

- MCS - <http://www.mcs.anl.gov/systems/software/>
- Scripts for synchronizing file systems between servers. Wrappers around `rdist` and `rsync`.



SC99 Tutorial Page 195



Tools - Monitoring

Supermon

- ACL - <http://www.acl.lanl.gov/software>
- Supermon is a cluster status server that allows multiple users to view the total activity of the cluster and determine how programs are affecting and being affected by the cluster.

Superview

- ACL - <http://www.acl.lanl.gov/software>
- Superview is a Java Swing client that can attach to a Supermon, and display cluster status information graphically

BEC

- ACL - <http://www.acl.lanl.gov/software>
- Benchmarking Extreme Clusters (BEC) is a benchmarking framework that makes testing and comparing the results of message passing and other metrics simple.



SC99 Tutorial Page 196



Tools - Monitoring

Performance Data Repository

- NERSC - <http://www.nersc.gov/research/ftg/pcp/performance.html>
- NAS Benchmark and other performance data for cluster technologies

NAS Benchmarks

- NASA - <http://www.nas.nasa.gov/Software/NPB/>
- Well-known benchmarks for parallel computers



Tools - Communication

MPICH

- MCS - <http://www.mcs.anl.gov/mpi>
- An implementation of MPI

LAM

- University of Notre Dame - <http://www.mpi.nd.edu/lam/>
- Another implementation of MPI

M-VIA

- NERSC - <http://www.nersc.gov/research/bld>
- An implementation of Virtual Interface Architecture for Linux



Tools - Communication

MVICH

- NERSC - <http://www.nersc.gov/research/bld>
- An implementation of the MPICH ADI for VIA



SC99 Tutorial Page 199



Tools - Resource Management

- PBS
 - NAS & MRJ - <http://pbs.mrj.com>
 - The Portable Patch System for scheduling. See most of Chapter 8 for details.



SC99 Tutorial Page 200



Cfengine Examples (from ACL configs):

```
groups:
    # Myrinet devices
    PMDevExists      = ( "/usr/bin/test -c /dev/pmv" )
    MyriDevExists     = ( "/usr/bin/test -c /dev/myri0" )

[...]
files:
    any::
        # Cert advisories
        /bin/mount m=0755 o=root action=fixall
        /bin/umount m=0755 o=root action=fixall

[...]
((LBPMachines|PeakMachines).MyriDevExists)::
    /dev/myri0 o=root g=sys m=0666 action=fixplain

[...]
LBPMachines|PeakMachines::
    { /etc/sysconfig/network
      DeleteLinesMatching "FORWARD_IPV4=false"
      SetLine "FORWARD_IPV4=no"
      AppendIfNoLineMatching "^FORWARD_IPV4=no.*"
    }
```



Appendix B - References

- *In Search of Clusters, Second Edition* by Gregory Pfister. Prentice Hall.
- *How to Build a Beowulf: A Guide to the Implementation and Application of PC Clusters* by Thomas Sterling et al. MIT Press.
- *Using Mpi : Portable Parallel Programming With the Message-Passing Interface, Volumes 1 and 2.* William Gropp et al. MIT Press.
- *MPI: The Complete Reference. Volumes 1 and 2* by William Gropp et al. MIT Press.
- Cluster-Related Web Sites
 - <http://www.beowulf.org>
 - <http://www.extremelinux.org>
 - <http://www.beowulf-underground.org>
 - Sandia National Laboratory's Cplant (Pioneering Work in Scalable Clusters)
 - <http://www.cf.sandia.gov/cplant/>
 - <http://www.nersc.gov/research/tribble> (this tutorial)

